DTU Library

MCNPX-McStas Interface

Klinkby, Esben Bryndt; Schönfeldt, Troels; Batkov, Konstantin; Rantsiou, Emmanouela; Filges, Uwe; Panzner, Tobias; Willendrup, Peter Kjær

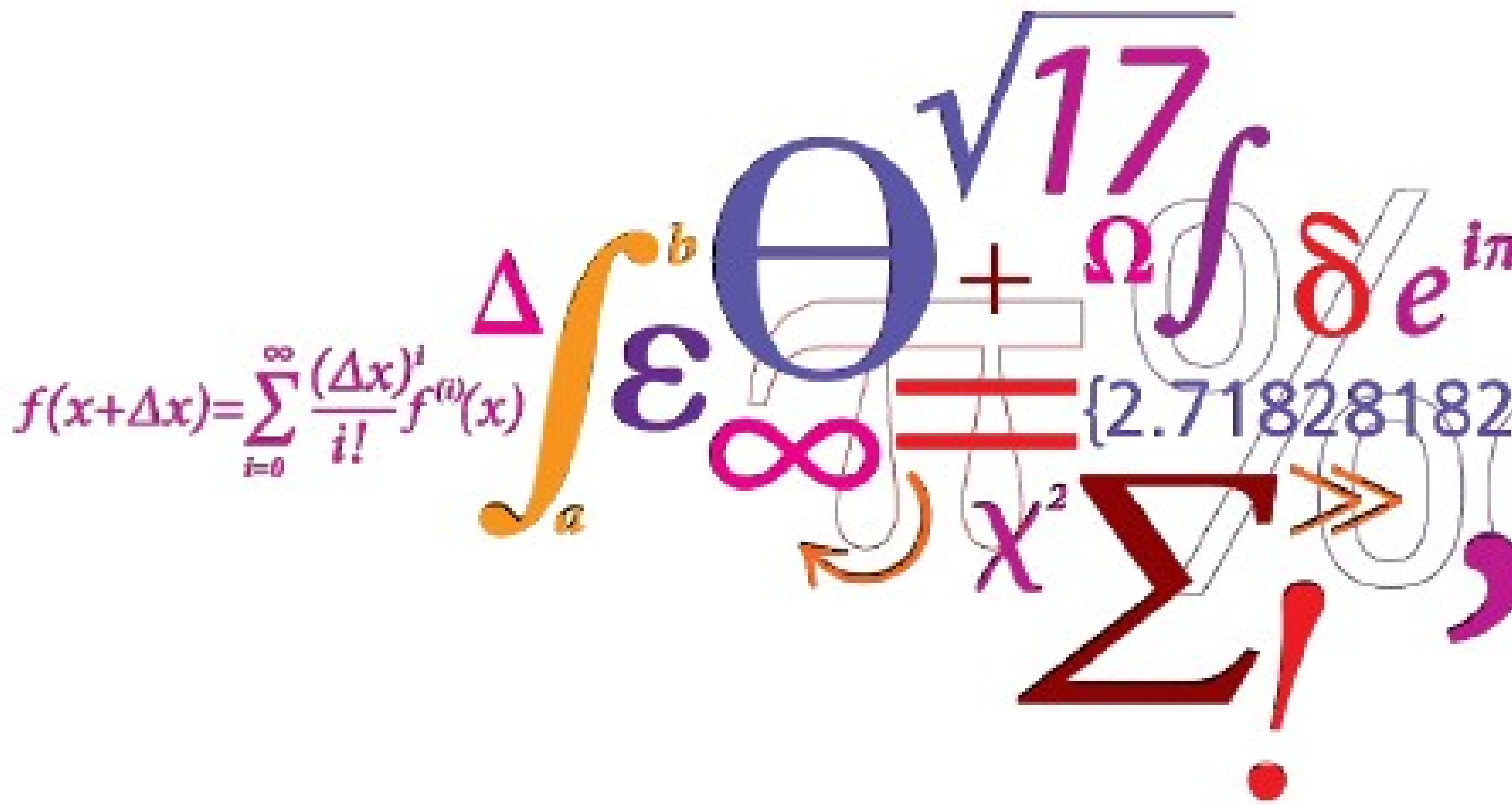
Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

MCNPX-McStas Interface

Esben Klinkby, DTU Nutech

» Collaboration with:
Troels Schönfeldt, Konstantin Batkov, Emmanouela Rantsiou, Uwe Filges, Tobias Panzner, Peter Willendrup, Erik Bergbäck and several others..

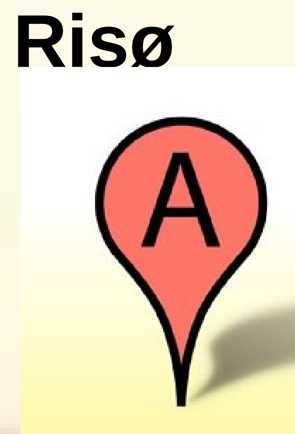


DTU Nutech
Center for Nukleare Teknologier

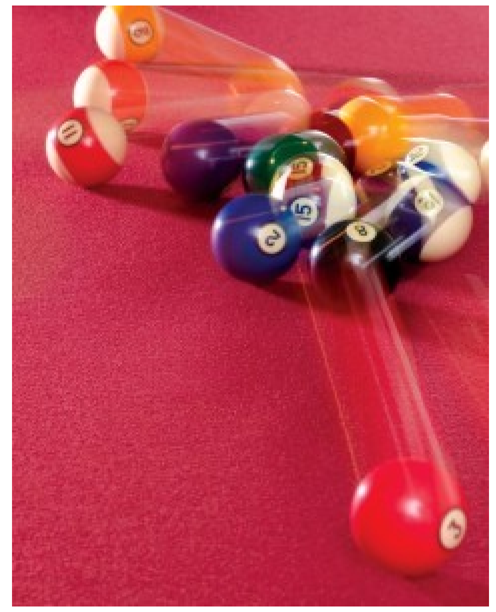
EUROPEAN
SPALLATION
SOURCE

Outline

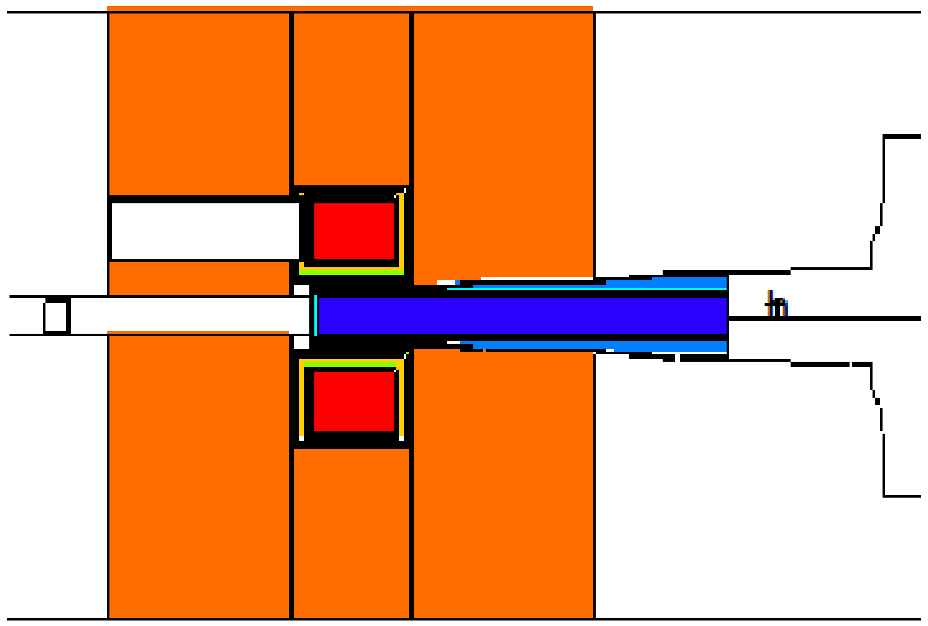
- Exploring possible interfaces
- Validation
- Examples of usage

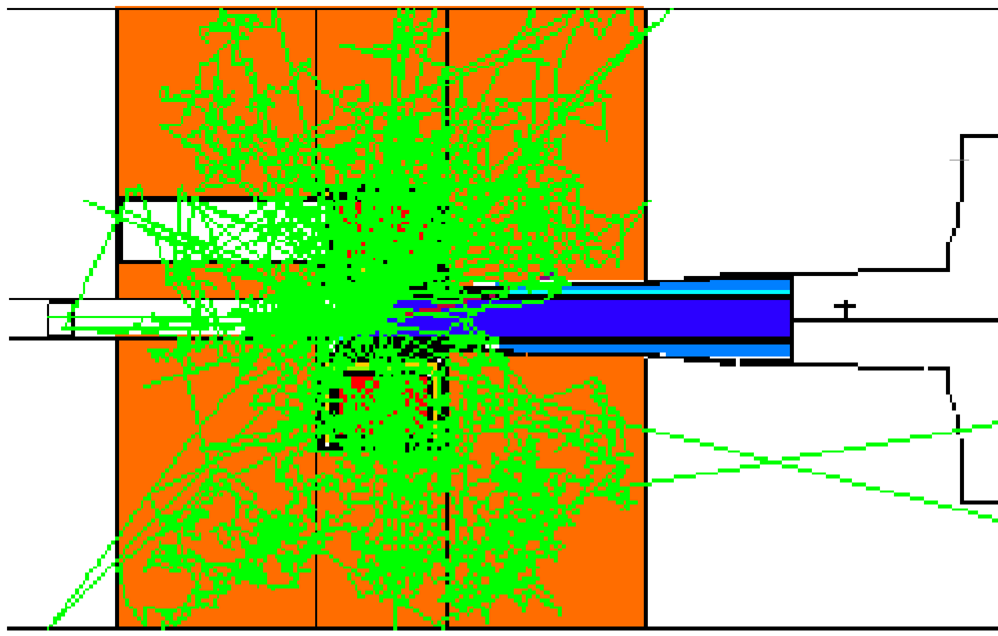


Neutron simulation



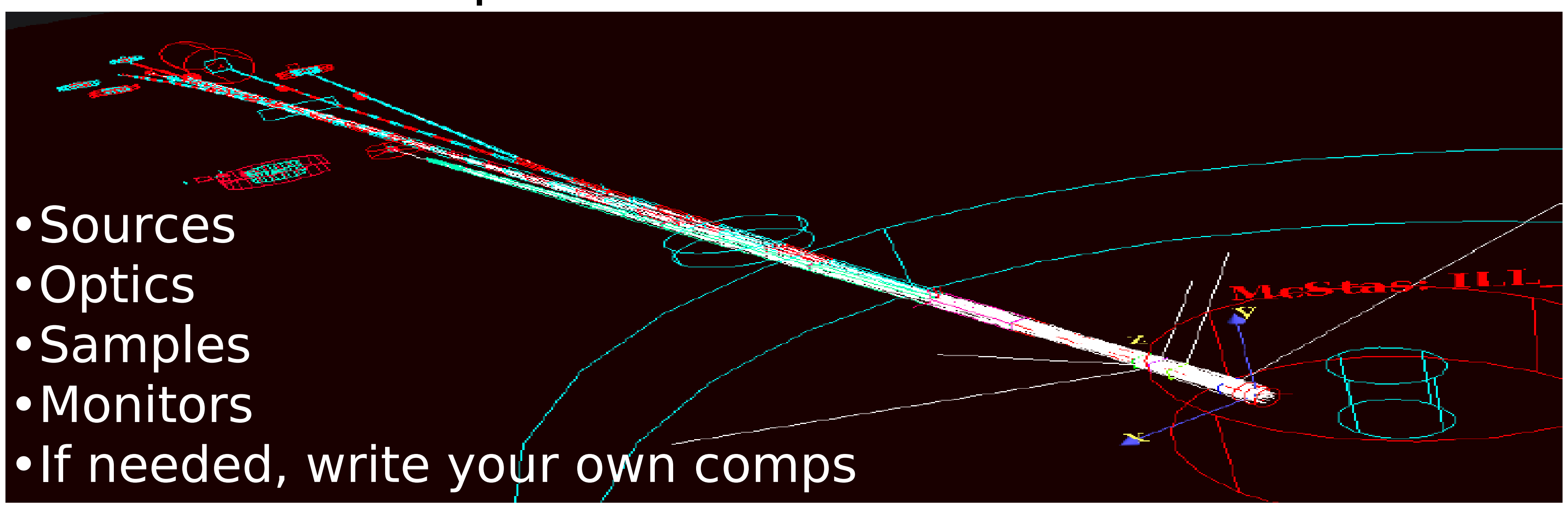
- Neutron production is well-described by:
  - **MCNPX** modeling: geometry, materials, cross-sections
  - Particle description: incoherent (billard balls)





Interaction of 1 proton
with the ESS target wheel 1!

- Neutron scatterers/instrument designers use ray-tracing codes:
  - **McStas**: geometry, optics
  - Wave description: coherent



Example from ILL
E.Farhi

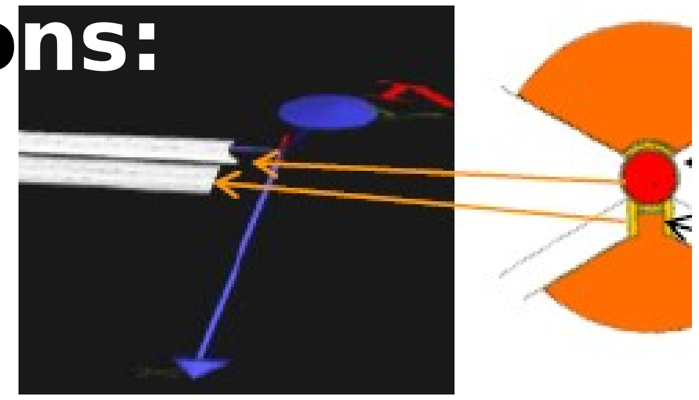
- Our task: interface the two simulation suites

The task:

“Interfacing the MCNP and McStas Monte Carlo codes for improved optimization of the ESS moderator-beam extraction systems”

The solutions:

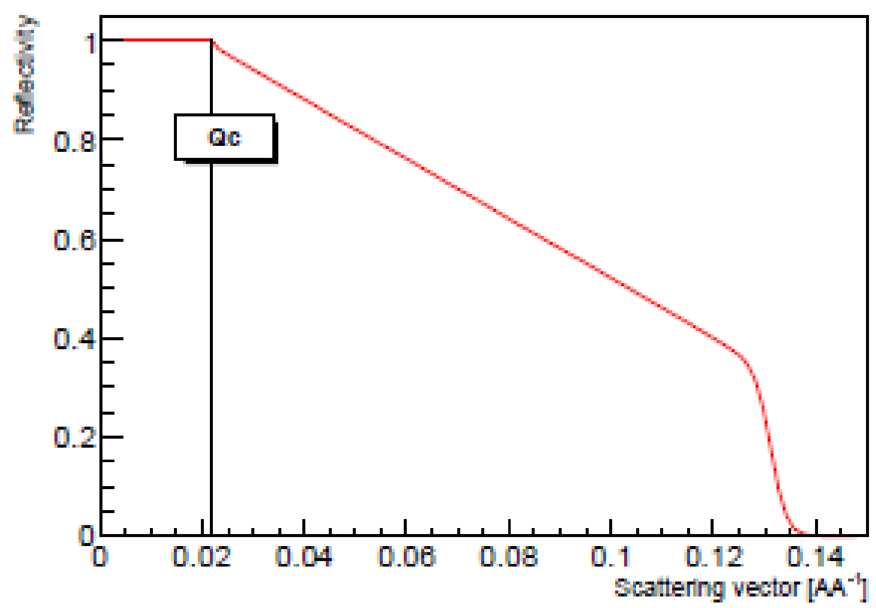
Tally



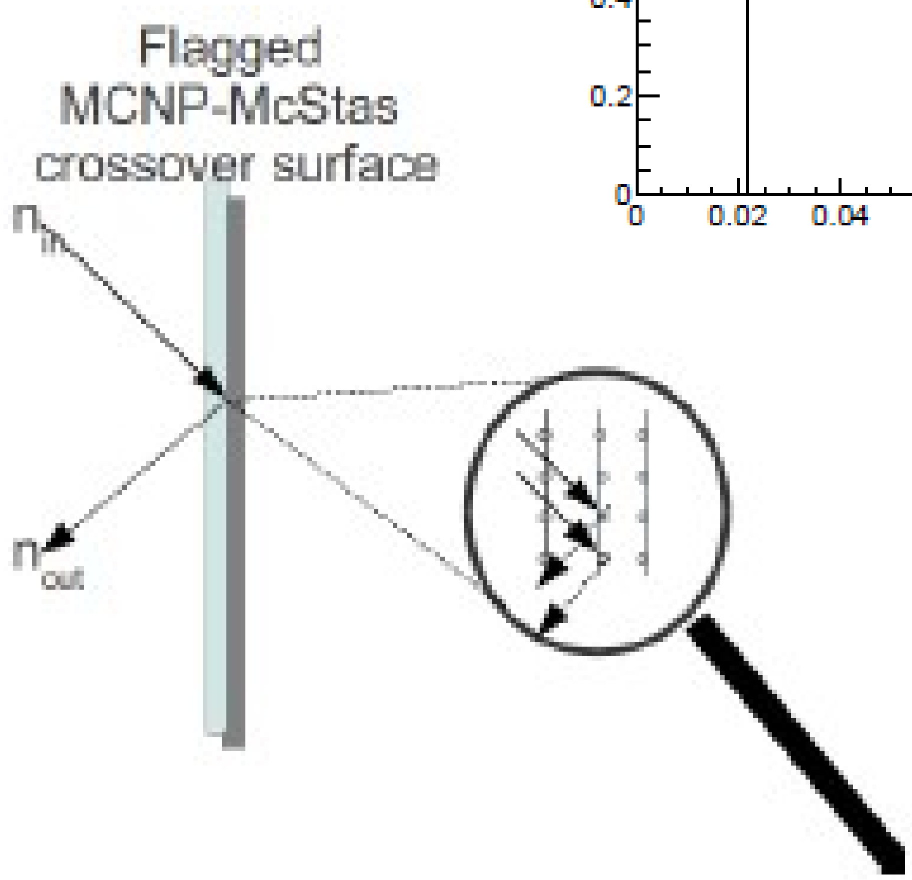
Ptrac

```
.........................................
 3000        2        10       179
100         2         0
   0.00000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.33356E-02
```

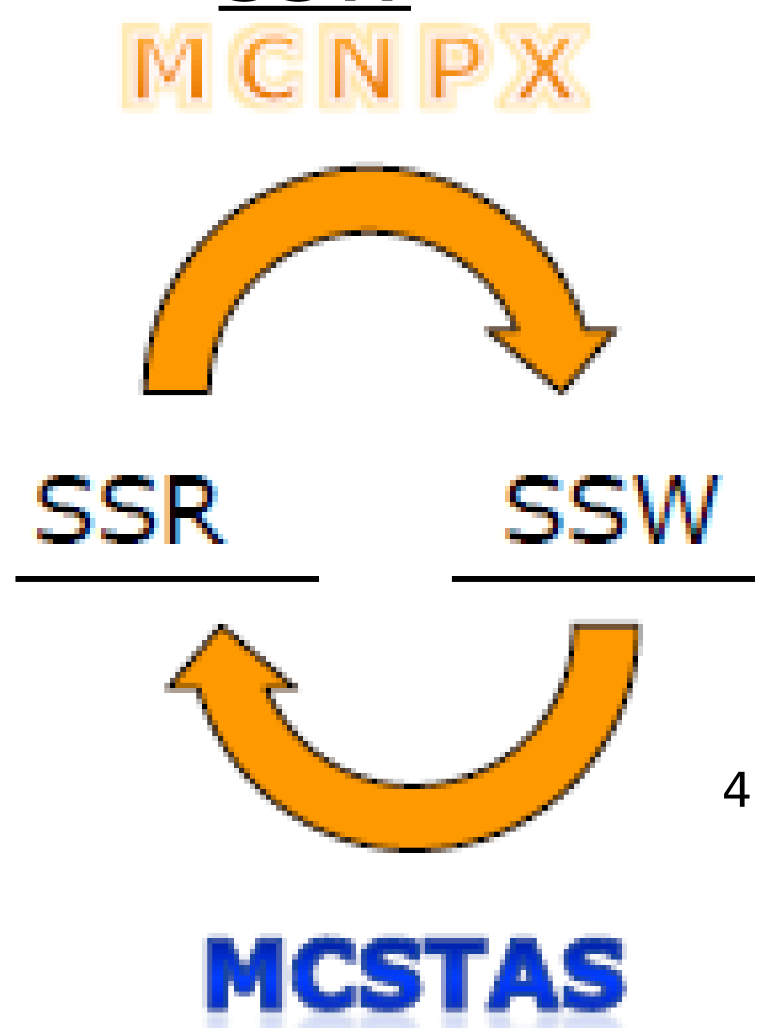
Supermirror



Compile



SSW

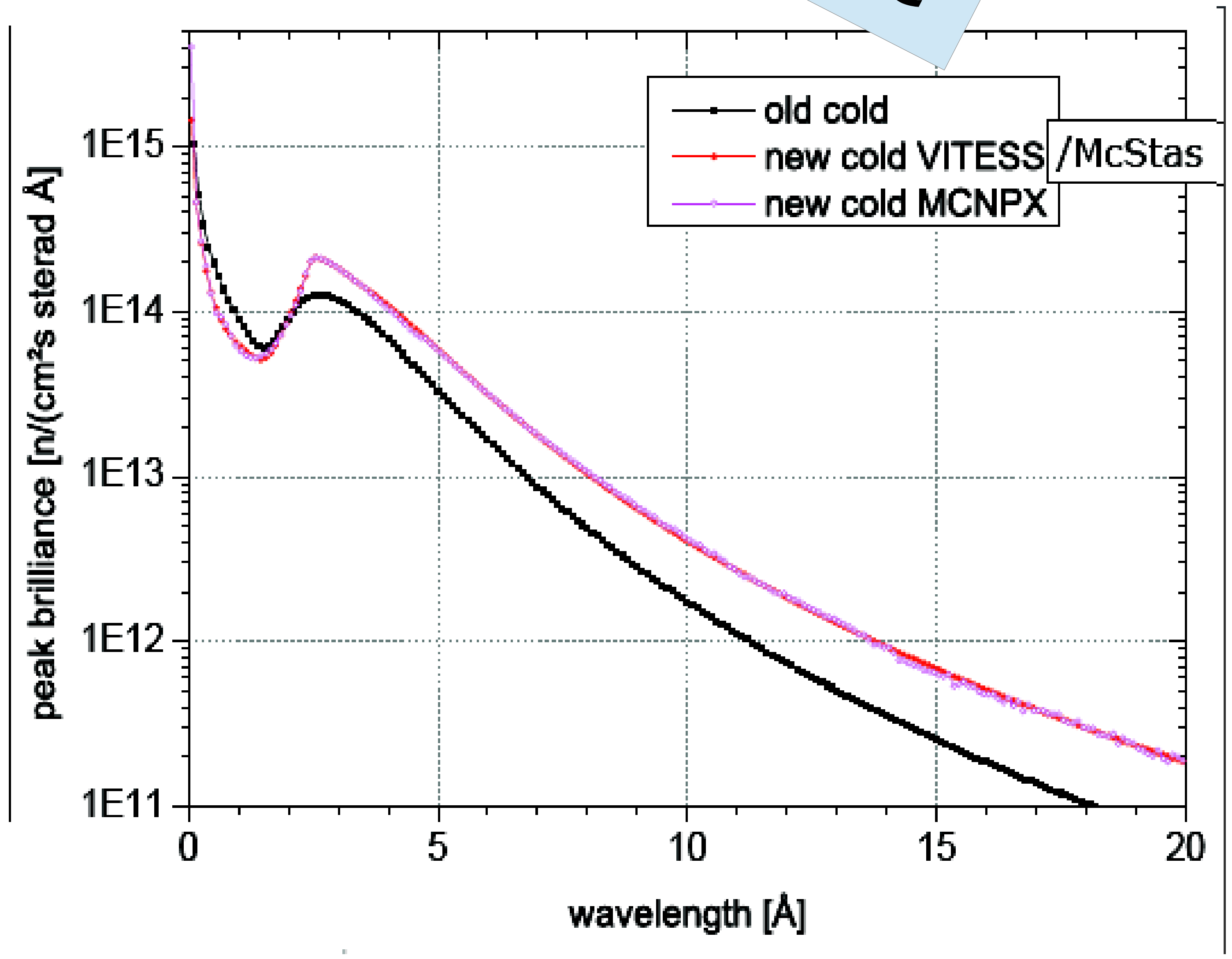


4

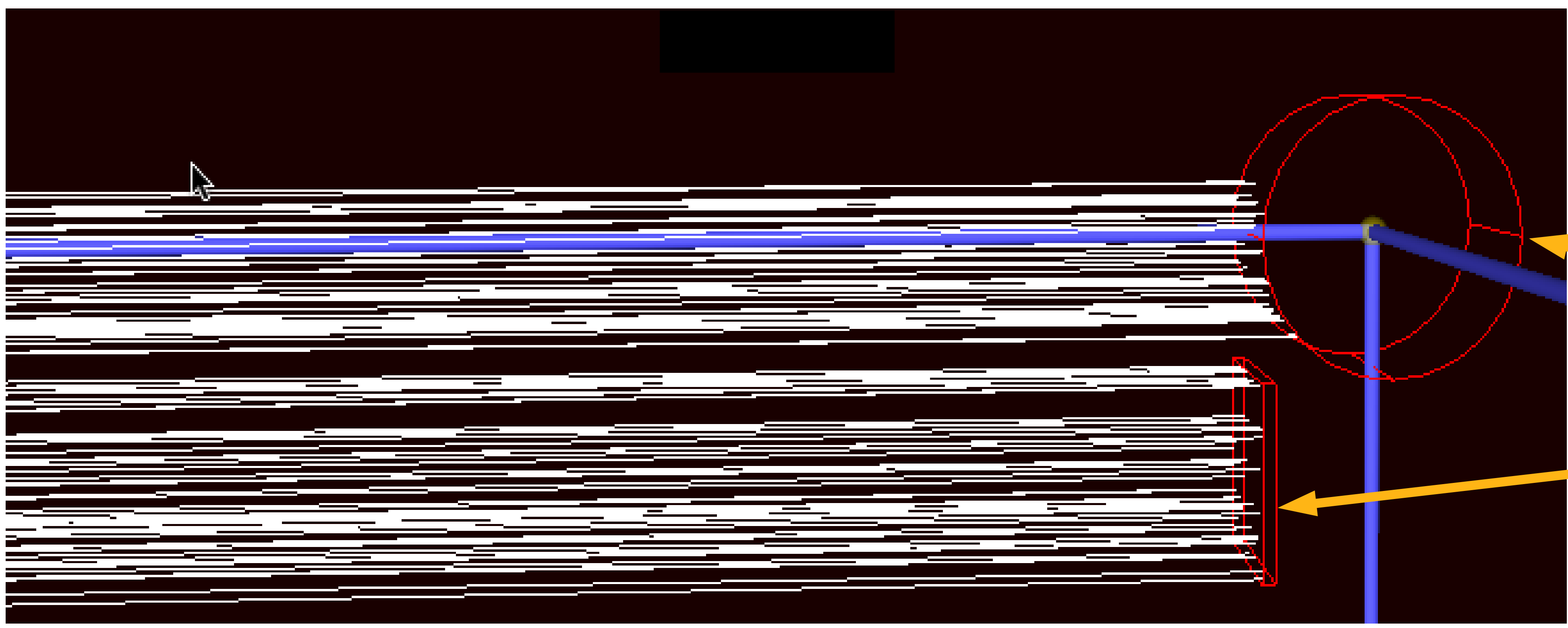
Interface option : **Tally fitting**

McStas Update

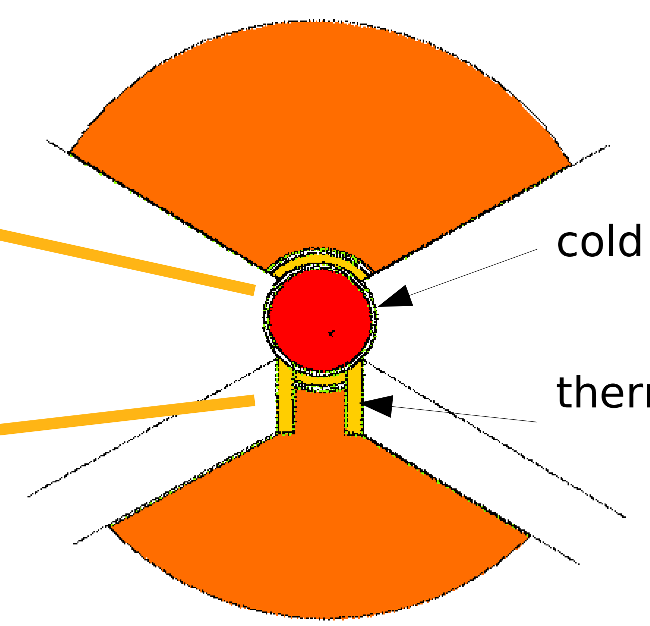
- Based on the latest MCNPX ESS target station (bi-spectral) geometry from ESS-Bilbao we have updated the McStas ESS source component mimicking both geometry and spectra.
- *ESS_Moderator_long*, and is part of McStas 2.0



McStas



MCNPX



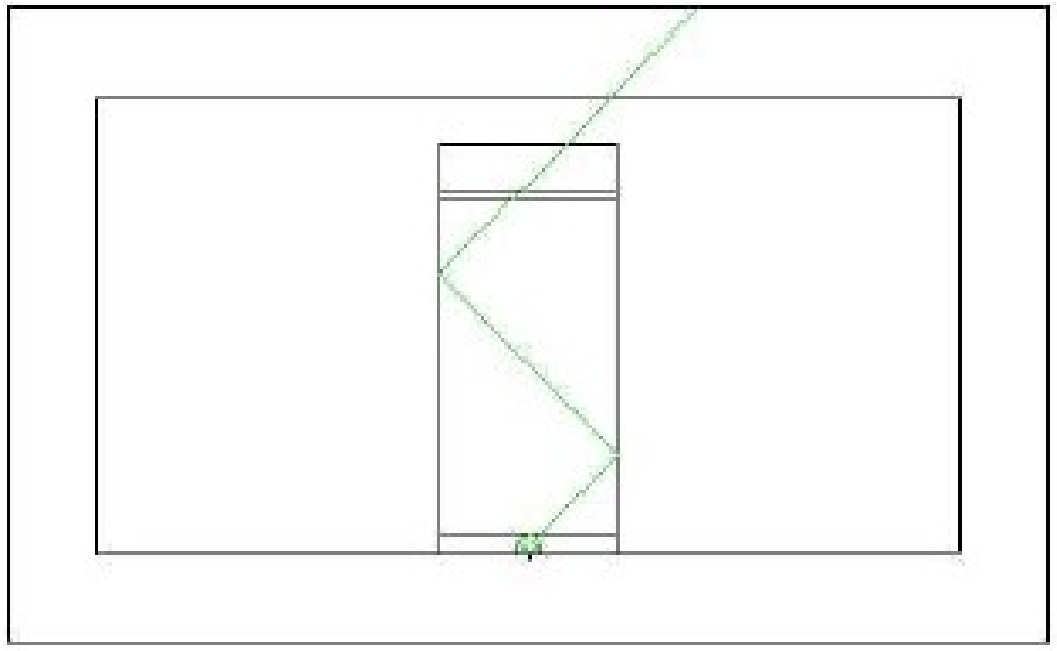
Cross comparison

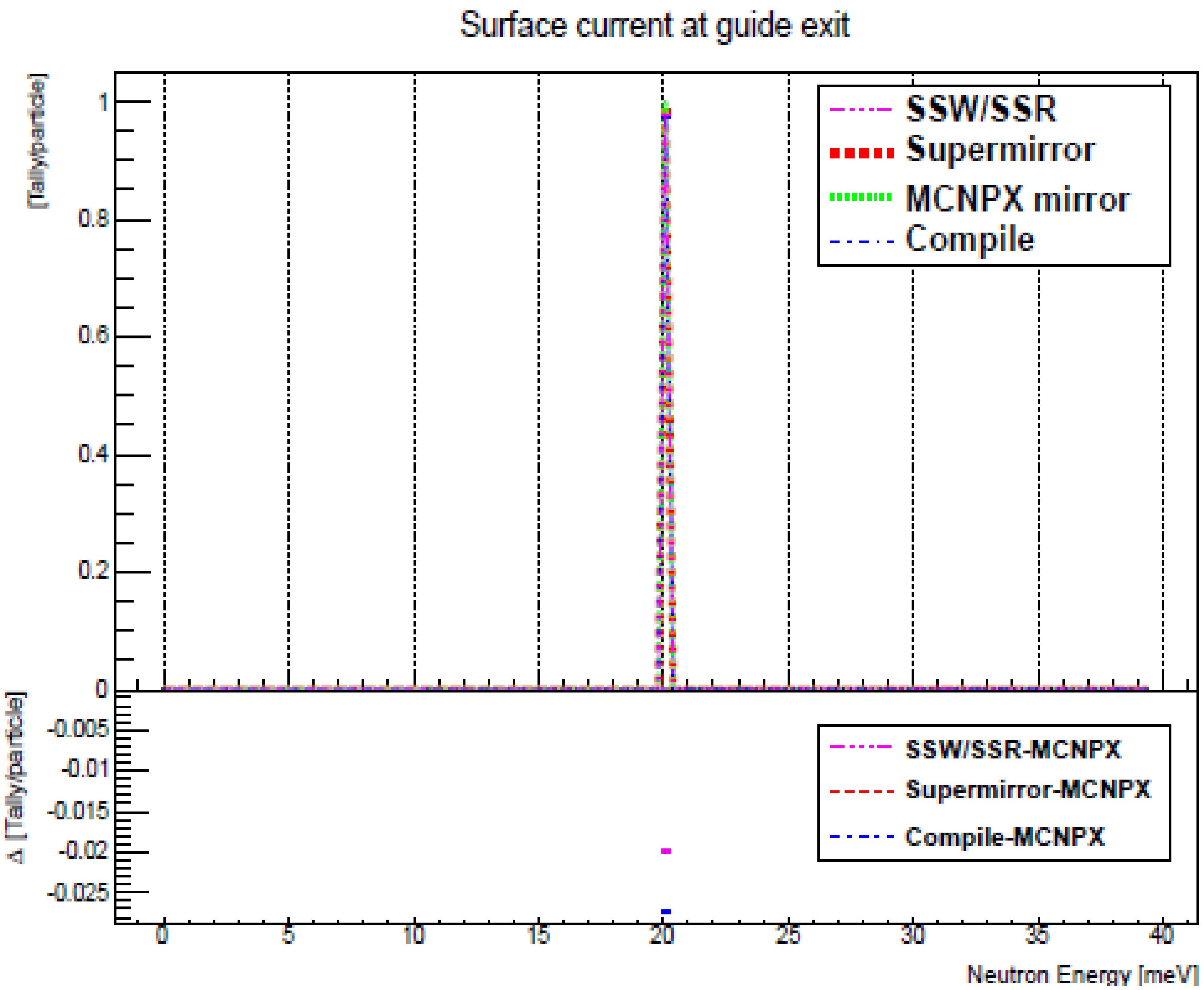
	Re-entry neutrons	Speed	Single neutron trace	Require License	Comments
Tally	No	Fast*	No	No	Should try to determine validity at least once
Ptrac	No	Fast*	Yes	Yes/No	Somewhat outdated by SSW/SSR
SSW/SSR	Yes	Fast*	Yes	Yes	Works well
Compile	Yes	Very slow	Yes	Yes	Require (minor) changes to MCNPX source code
Supermirror	Yes	Slow	yes	yes	Generalizes poorly (but who cares?)

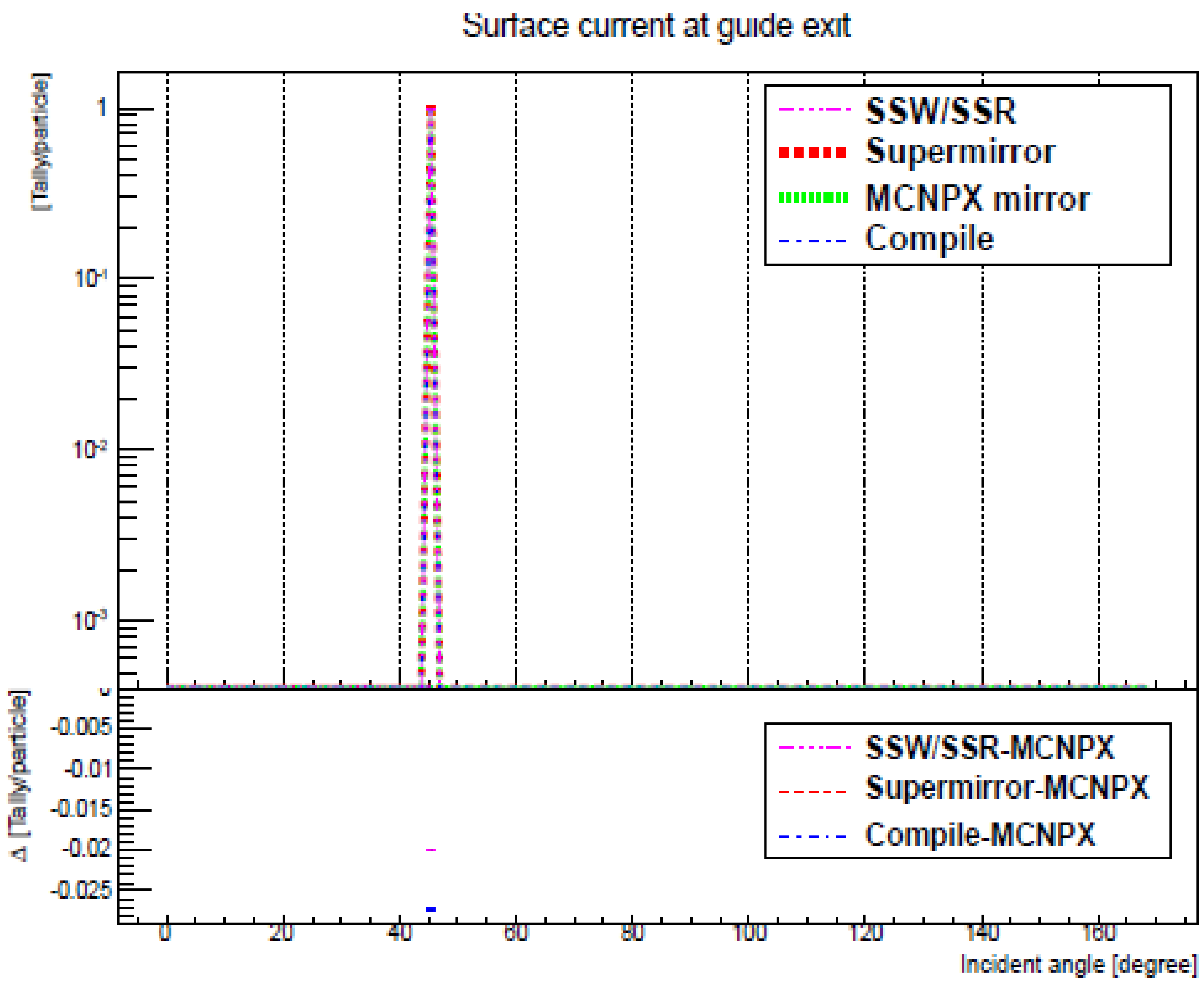
*) The computational heavy MCNP/X calculation can be performed once-and-for-all

Validation results

- ➢ Software validated on dummy setup: fire undivergent, monoenergetic neutrons at a perfectly reflecting guide wall:



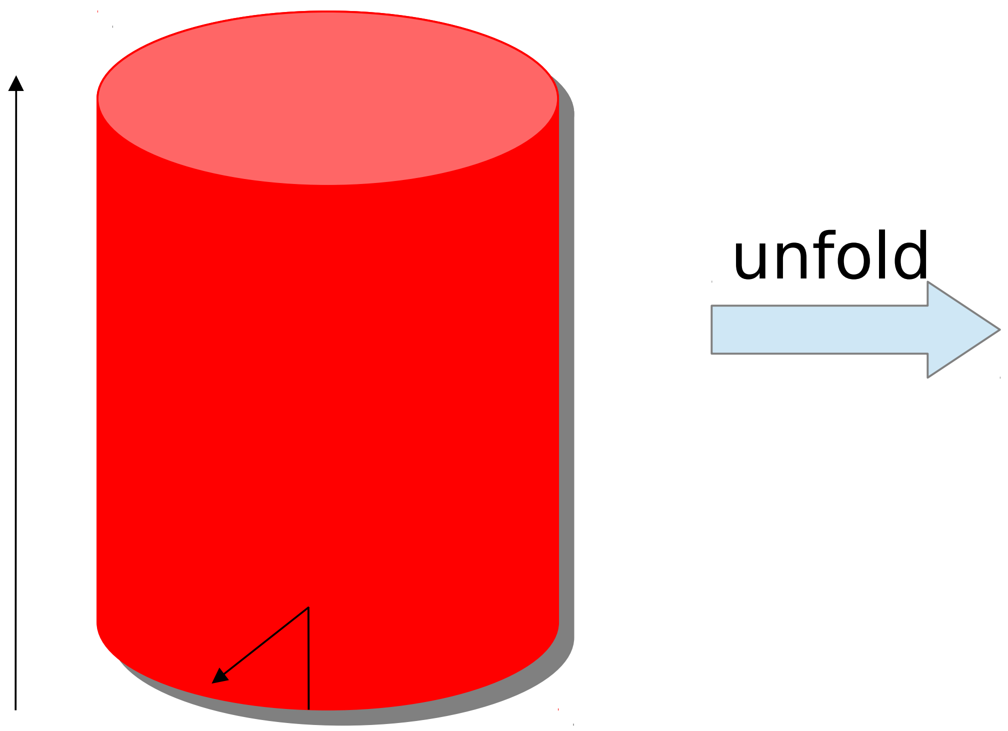


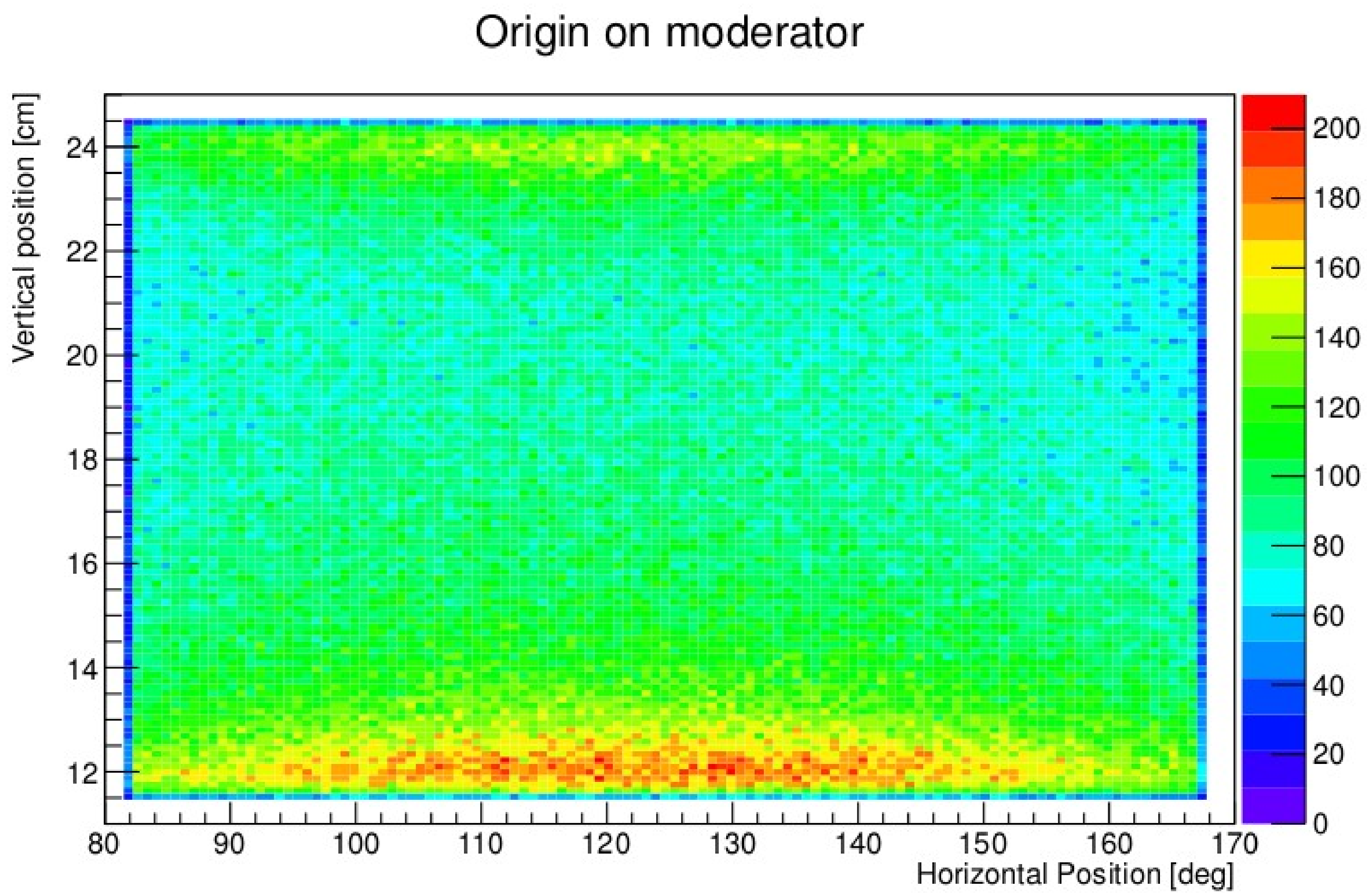


For all interfaces:
→ Neutron energy and angle conserved (45°, scattered twice)

Interface applications: neutron yield

- Software developed to transfer the individual neutron states (**r**,**v**,E,t,weight) from MCNPX to McStas/ROOT
- Avoids questionable phase-space assumptions





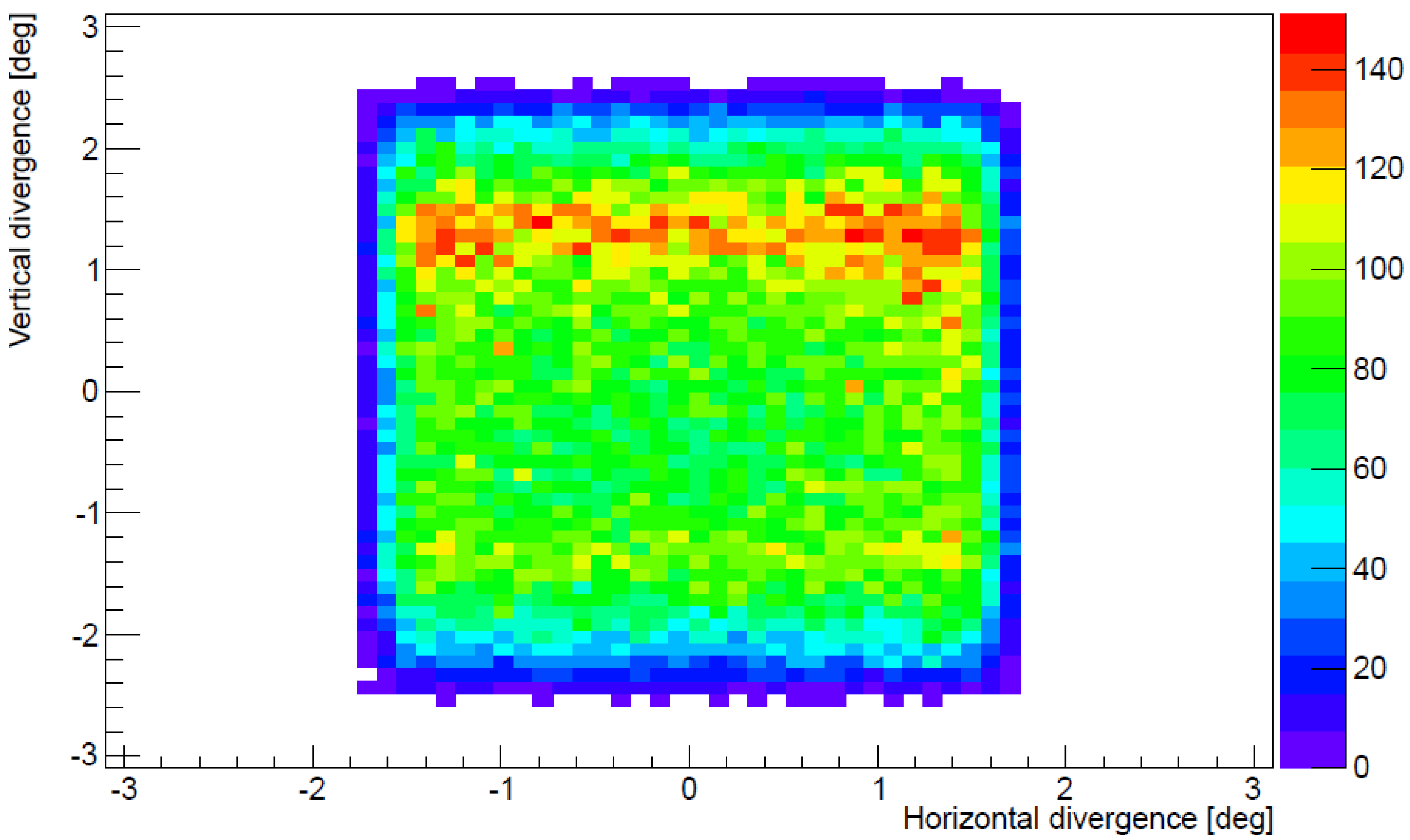
- A detailed look at the moderator, shows up to a factor ~2 difference in flux, on the surface (+differences in: divergence, spectrum, time)

Interface applications: placing instruments

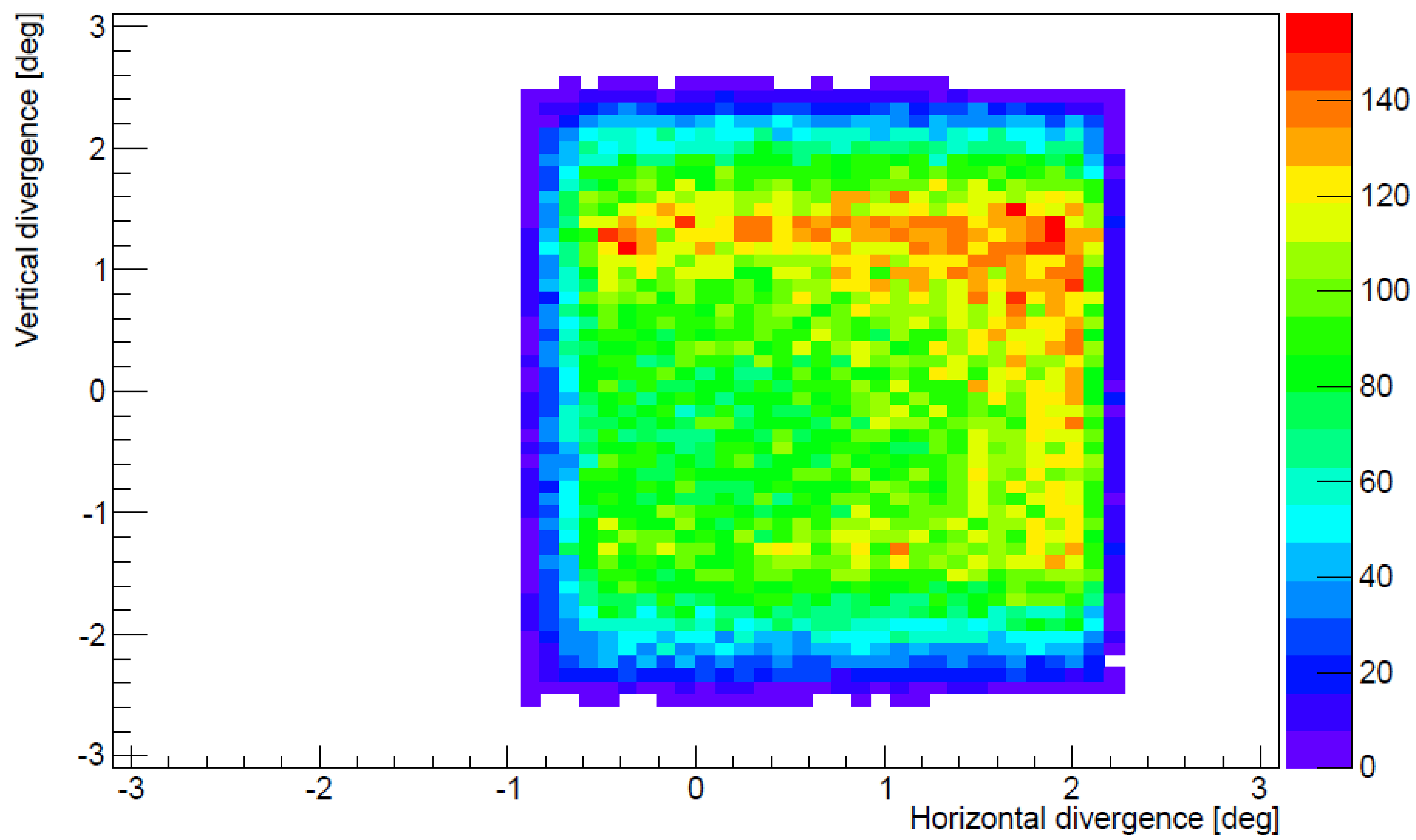
MCNPX simulation of proton bunches interactions ⇒ neutrons emitted from the cold moderator surface handed to interface and visualized

- Moderator is not uniform ⇒ beamport + guide orientation matters

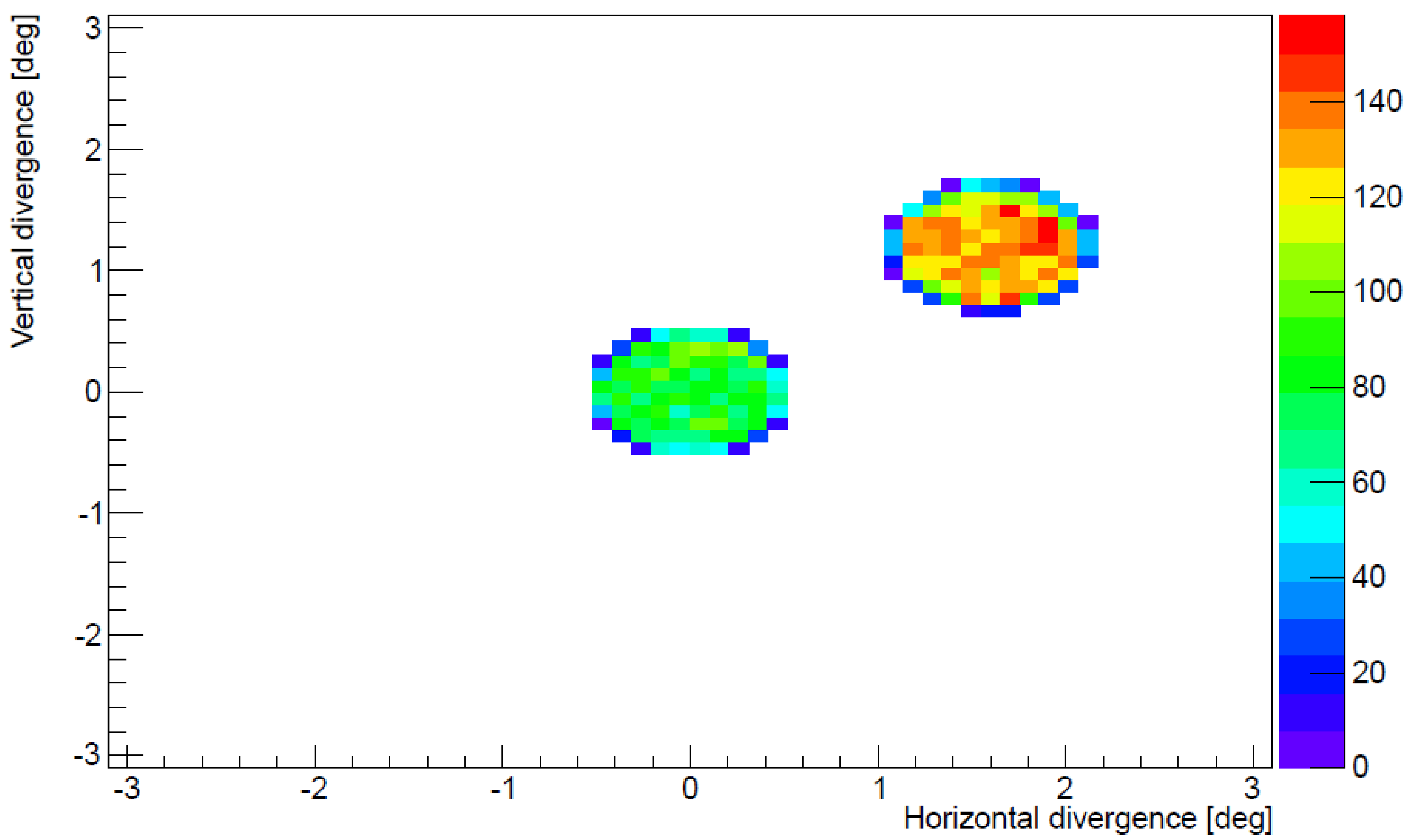
Central instrument



Left side instrument



Neutron guide acceptance (direct and angled view)

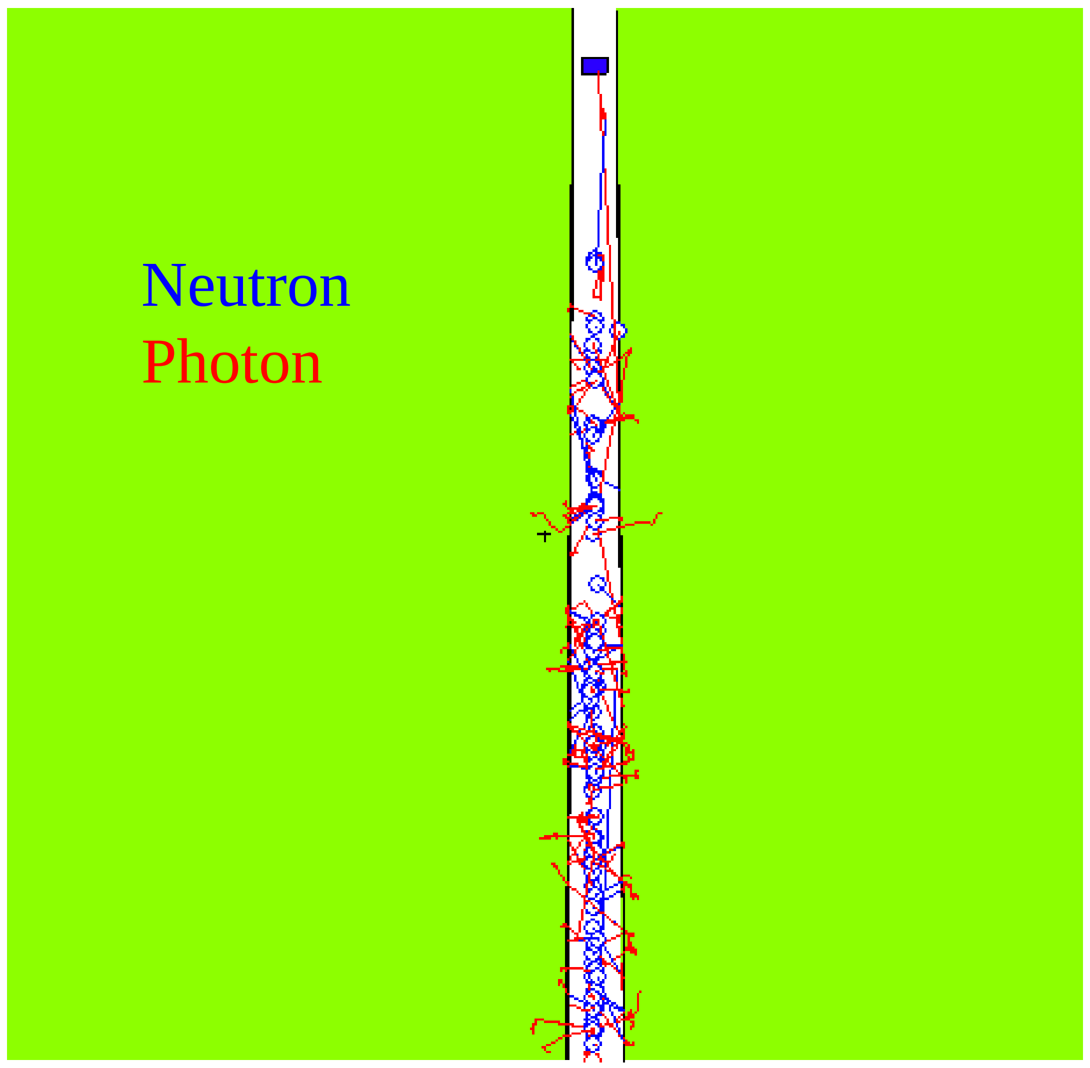


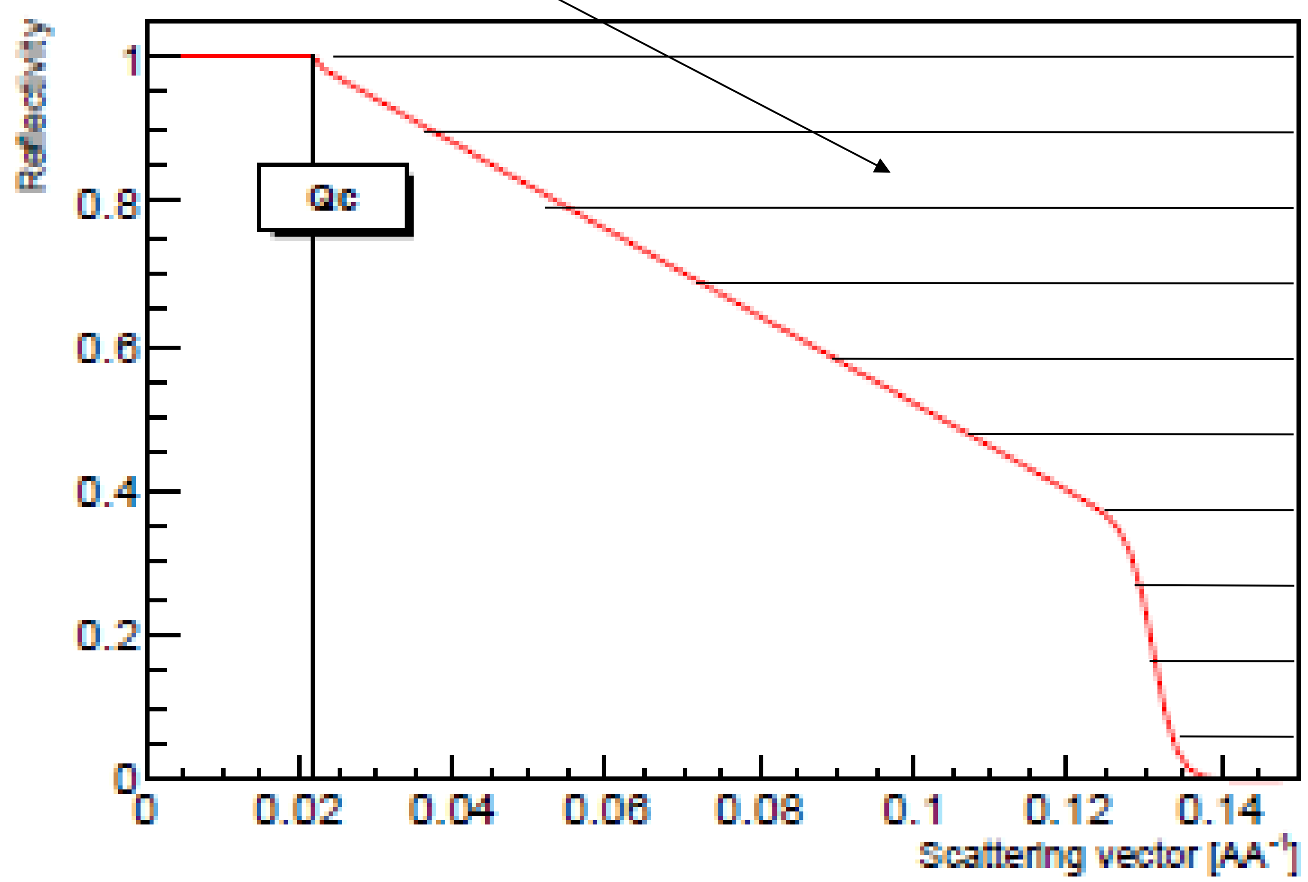
- Lesson learned: If the instrument considered can cope with the divergence penalty: Aim at the moderator with an angle and win up to ~50% neutron intensity

Interface applications: Background along guide

- Interface supports re-entry. i.e.
  - MCNPX → McStas → MCNPX

Example: Simulation of neutrons interactions in a guide
- Per default: McStas disregards unreflected neutrons
- Per default: MCNPX doesn't handle reflections

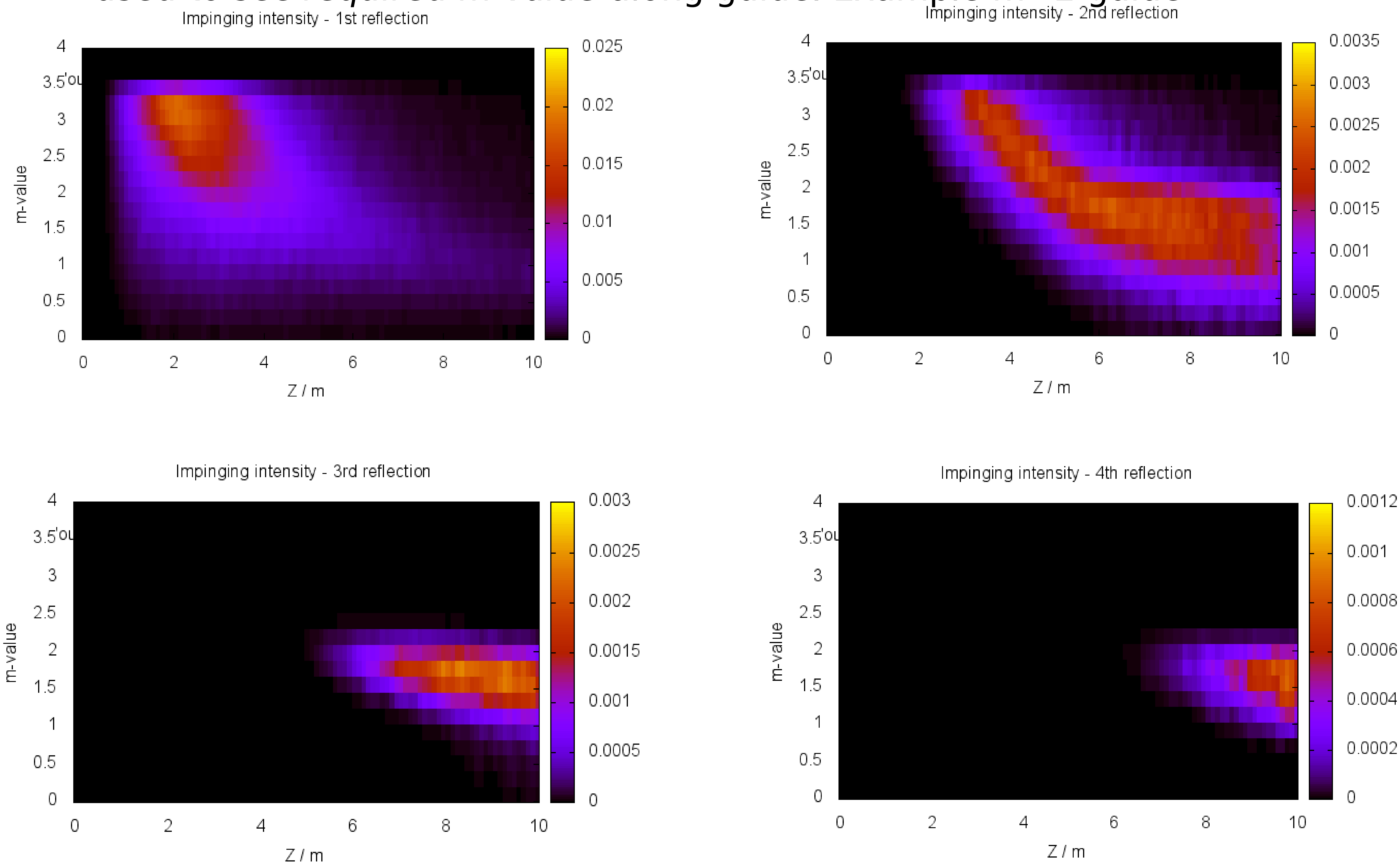




- Ongoing: Exploit to calculate shielding requirements along guide

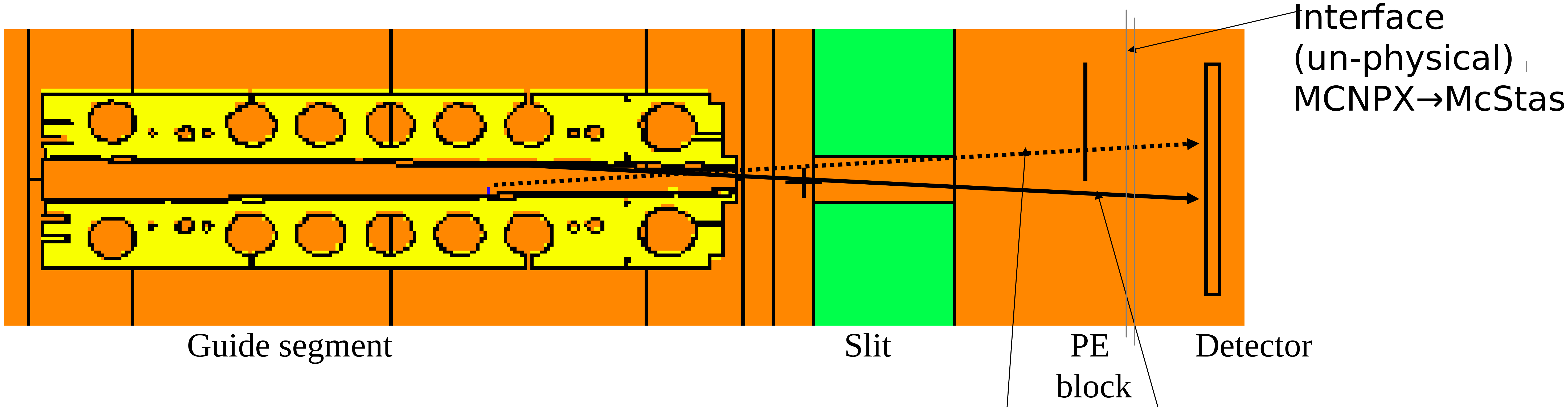
Interface applications: Background along guide (related)

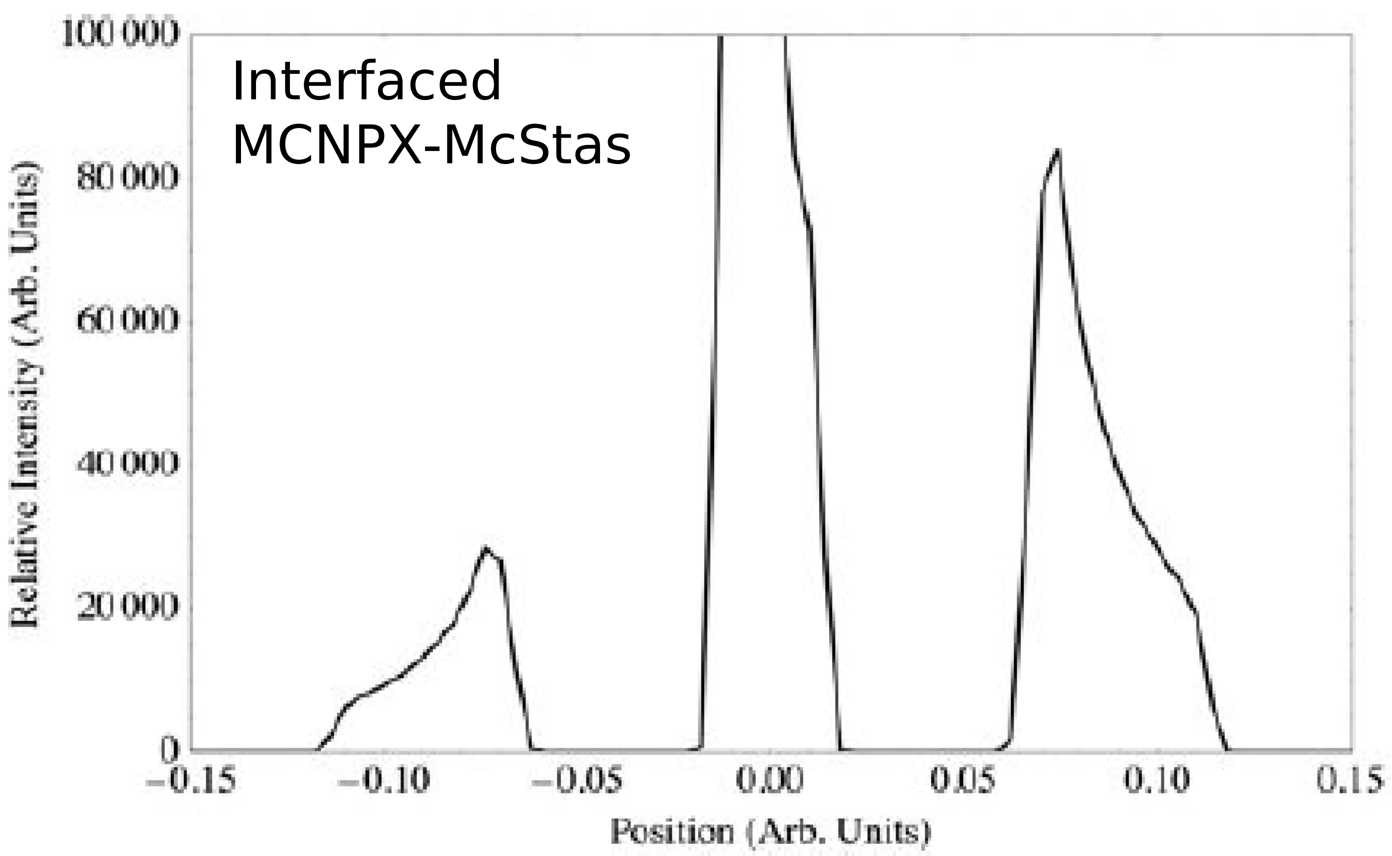
- The “logging mechanism” developed and used for gamma background, can also be used to see *required m-value* along guide. Example m=2 guide

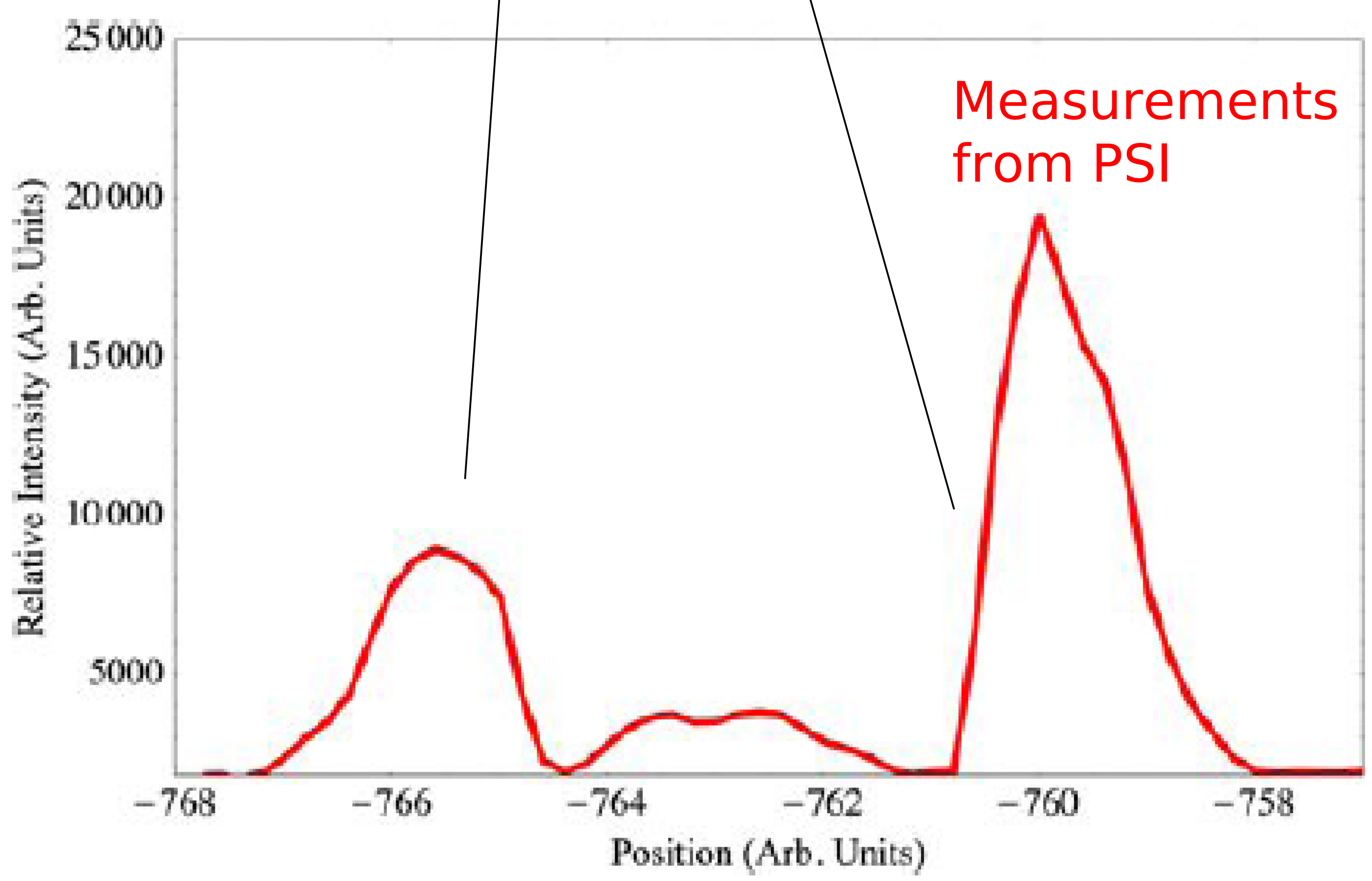


- Of course, results depends heavily on source & guide description

Interface applications: Downstream material







Per default: McStas does not handle material effects
Per default: MCNPX does not handle supermirrors
→ The combination using SSW/SSR describes the observations well

MCNPX-McStas interface paper

Nuclear Instruments and Methods in Physics Research A 700 (2013) 106–110

Contents lists available at SciVerse ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Interfacing MCNPX and McStas for simulation of neutron transport

Esben Klinkby [a,c,*], Bent Lauritzen [a,c], Erik Nonbøl [a,c], Peter Kjær Willendrup [b,c], Uwe Filges [d,e], Michael Wohlmuther [d,e], Franz X. Gallmeier [f]

[a] DTU Nutech, Technical University of Denmark, DTU Risø Campus, Frederiksborgvej 399, DK-4000 Roskilde, Denmark
[b] DTU Physics, Technical University of Denmark, DTU Lyngby Campus, Anker Engelunds Vej 1, DK-2800 Kgs. Lyngby, Denmark
[c] ESS Design Update Programme, Denmark
[d] Paul Scherrer Institute, CH-5232 Villigen PSI, Switzerland
[e] ESS Design Update Programme, Switzerland
[f] Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

A R T I C L E I N F O

Article history:
Received 29 June 2012
Received in revised form
31 August 2012
Accepted 15 October 2012
Available online 23 October 2012

Keywords:
Neutron
Transport
Simulation
MCNPX
McStas
Interface

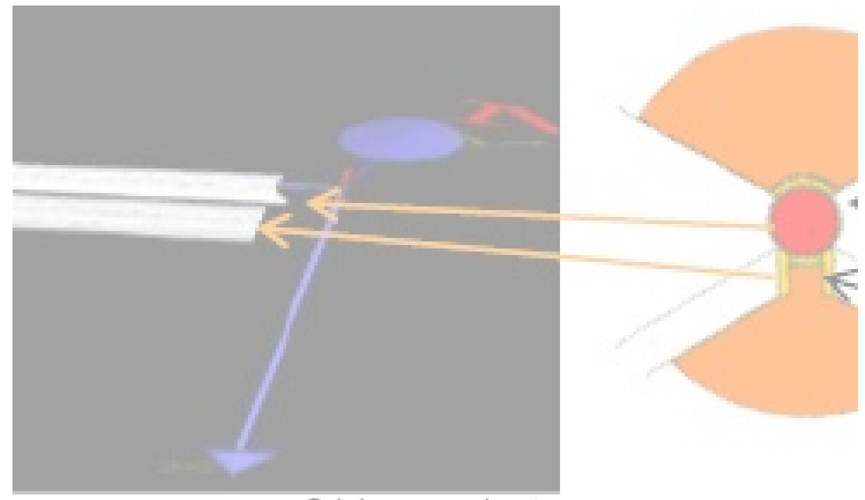
A B S T R A C T

Simulations of target-moderator-reflector system at spallation sources are conventionally carried out using Monte Carlo codes such as MCNPX (Waters et al., 2007 [1]) or FLUKA (Battistoni et al., 2007; Ferrari et al., 2005 [2,3]) whereas simulations of neutron transport from the moderator and the instrument response are performed by neutron ray tracing codes such as McStas (Lefmann and Nielsen, 1999; Willendrup et al., 2004, 2011a,b [4–7]). The coupling between the two simulation suites typically consists of providing analytical fits of MCNPX neutron spectra to McStas. This method is generally successful but has limitations, as it e.g. does not allow for re-entry of neutrons into the MCNPX regime. Previous work to resolve such shortcomings includes the introduction of McStas inspired supermirrors in MCNPX. In the present paper different approaches to interface MCNPX and McStas are presented and applied to a simple test case. The direct coupling between MCNPX and McStas allows for more accurate simulations of e.g. complex moderator geometries, backgrounds, interference between beam-lines as well as shielding requirements along the neutron guides.

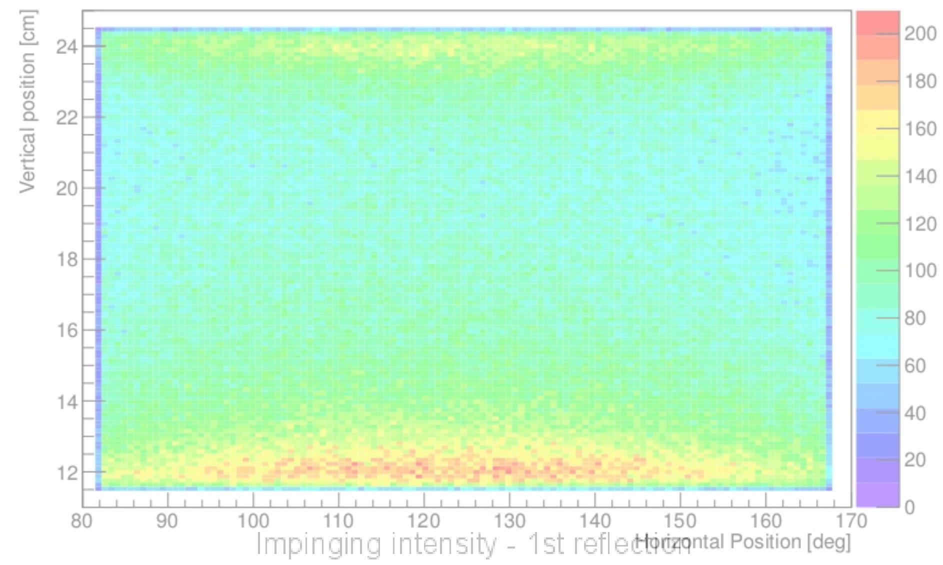
http://dx.doi.org/10.1016/j.nima.2012.10.052

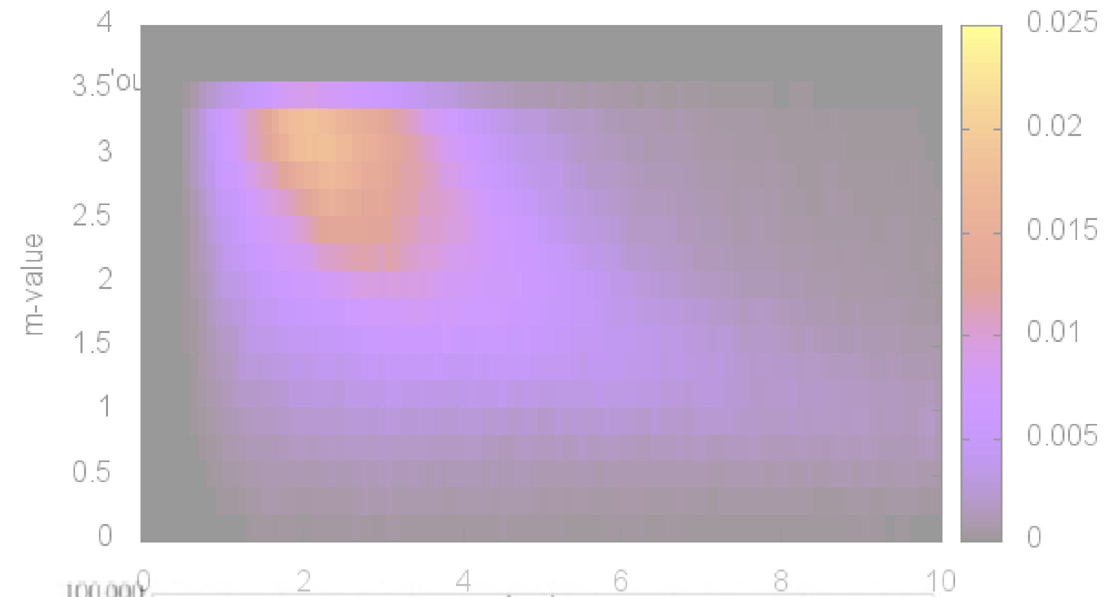
Conclusions & outlook

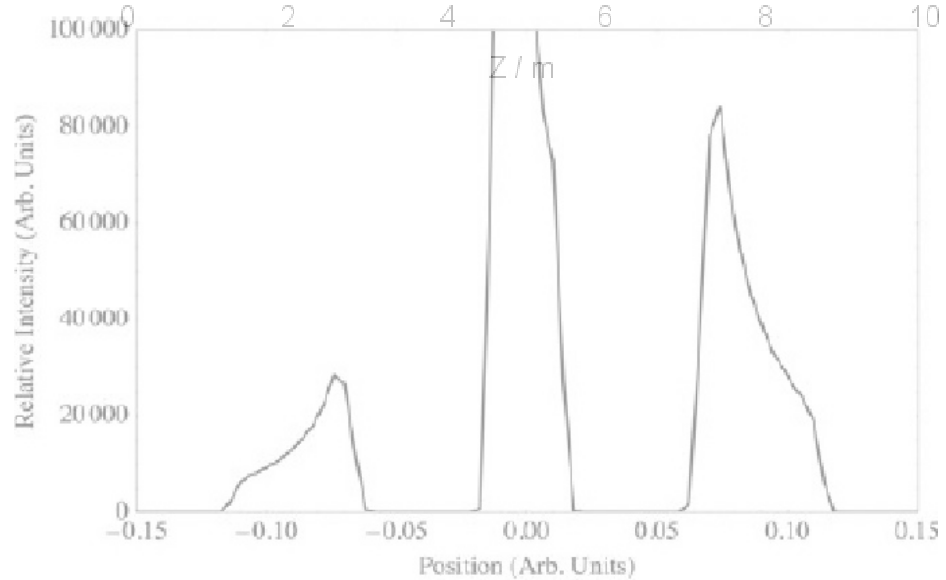
- Possible interfaces between MCNPX and McStas have been studied and evaluated
- The SSW is particularly useful → software is written and validated to communicate with McStas/ROOT through this interface
- The SSW interface has been applied to a number of use-cases:
  - Positioning instruments
  - Background along guide
  - Downstream material
  - Adaptive optics
- Useful tool, many applications in the future
- Interested, and didn't find sufficient info?:
  - Please don't hesitate to write a mail!

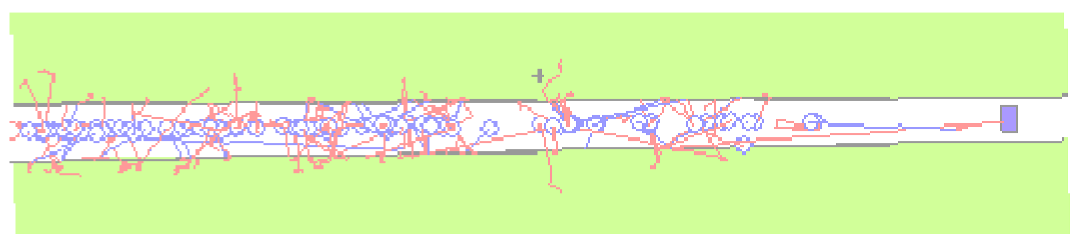


Origin on moderator





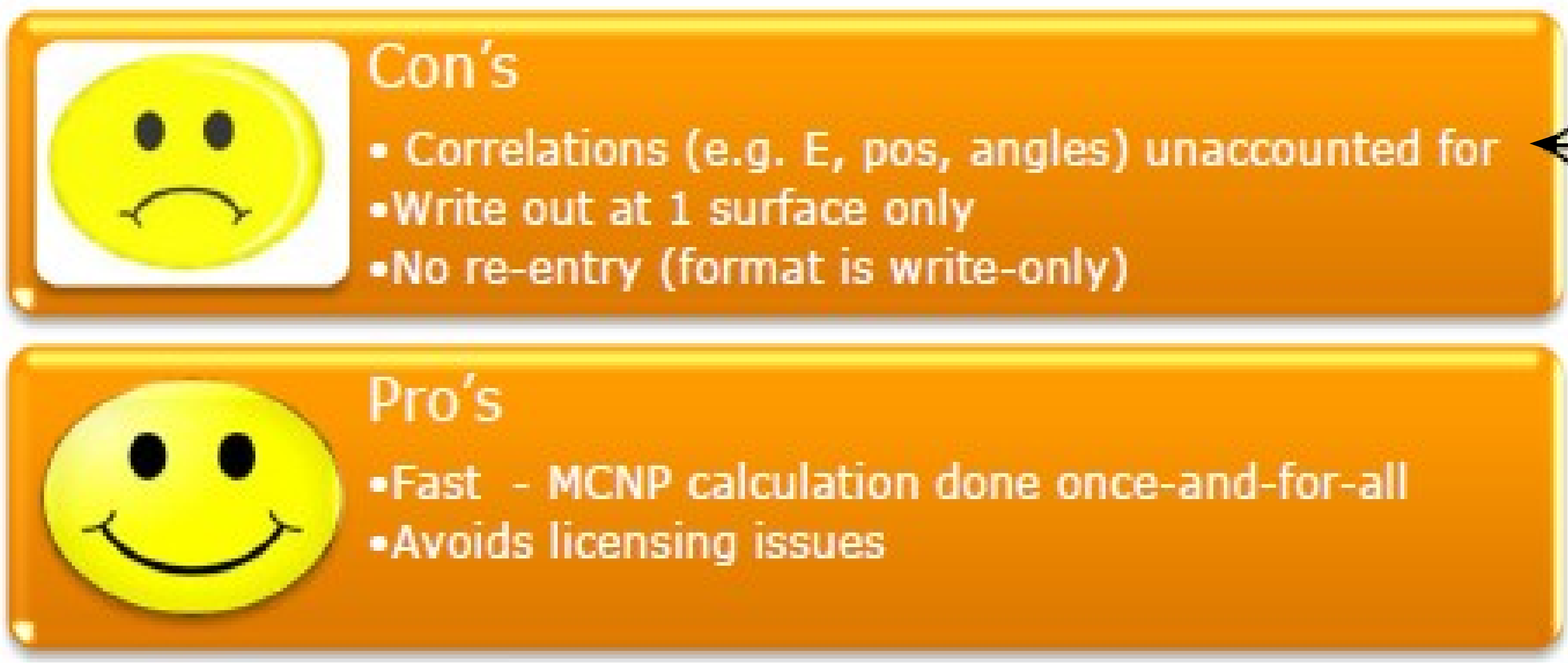




Backup slides

Interface option : **Tally fitting** (present default approach)

1. Neutron spectrum calculated with MCNP/X at the moderator surface
2. Spectrum is approximated by simple functions which serves as input to McStas.



Discussed later

Interface option : **Ptrac**

- ➢MCNPX can output an ascii file containing individual neutron states: pos, angles, energy, time & weight
- ➢The McStas component: *MCNP_Virtual_Input* converts the neutron state into McStas readable and works as a source

Con's

- ascii file enormous: ~0.2kB/evt
- Write out at 1 surface only
- No re-entry (format is write-only)
- Cannot run MPI

Pro's

- Correlations conserved (e.g. E,pos)
- Fast

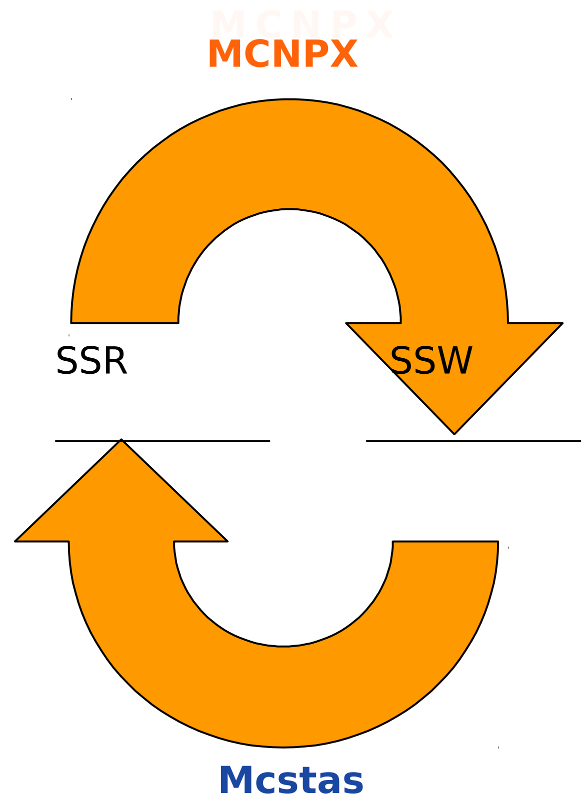
Ptrac format

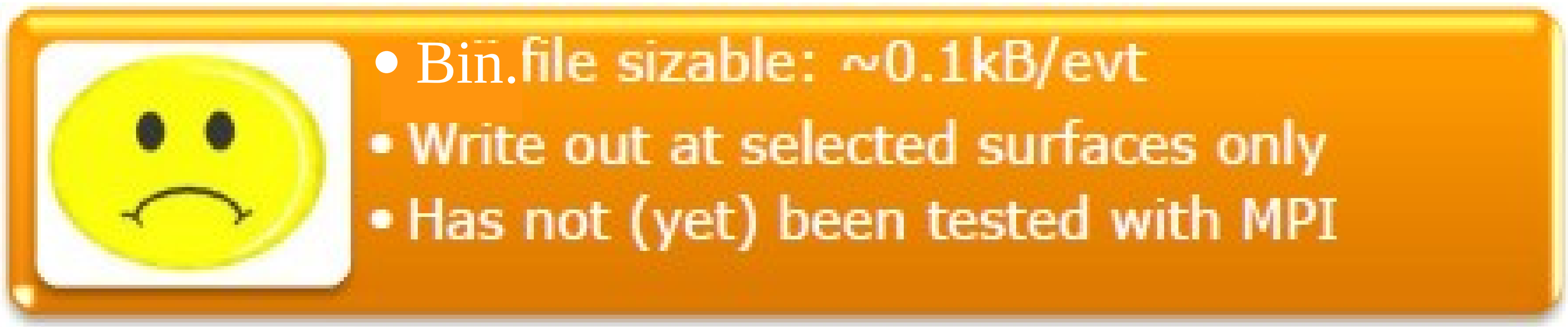
```
....................................
 3000       2      10      179      100
      2        0
   0.00000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.33356E-02
      3000       3      110      179
10        2        0
  -0.20000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.40028E-02
      3000       4      120      179
100        2        0
  -0.40000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.46699E-02
      3000       5      130      179
..................................
```

Interface option : **SSW/SSR**

- **S**ource **S**urface **R**ead/**W**rite in MCNPX starts/stops simulations at a given (set of) surface(s)
- The neutron state written to binary file.
- New McStas 2.0 components:

  MCNP_Virtual_ss_Input & *MCNP_Virtual_ss_Output* reads MCNPX output and writes MCNPX input
- Neutron propagation started in MCNPX, continued in McStas and finalizing in MCNP

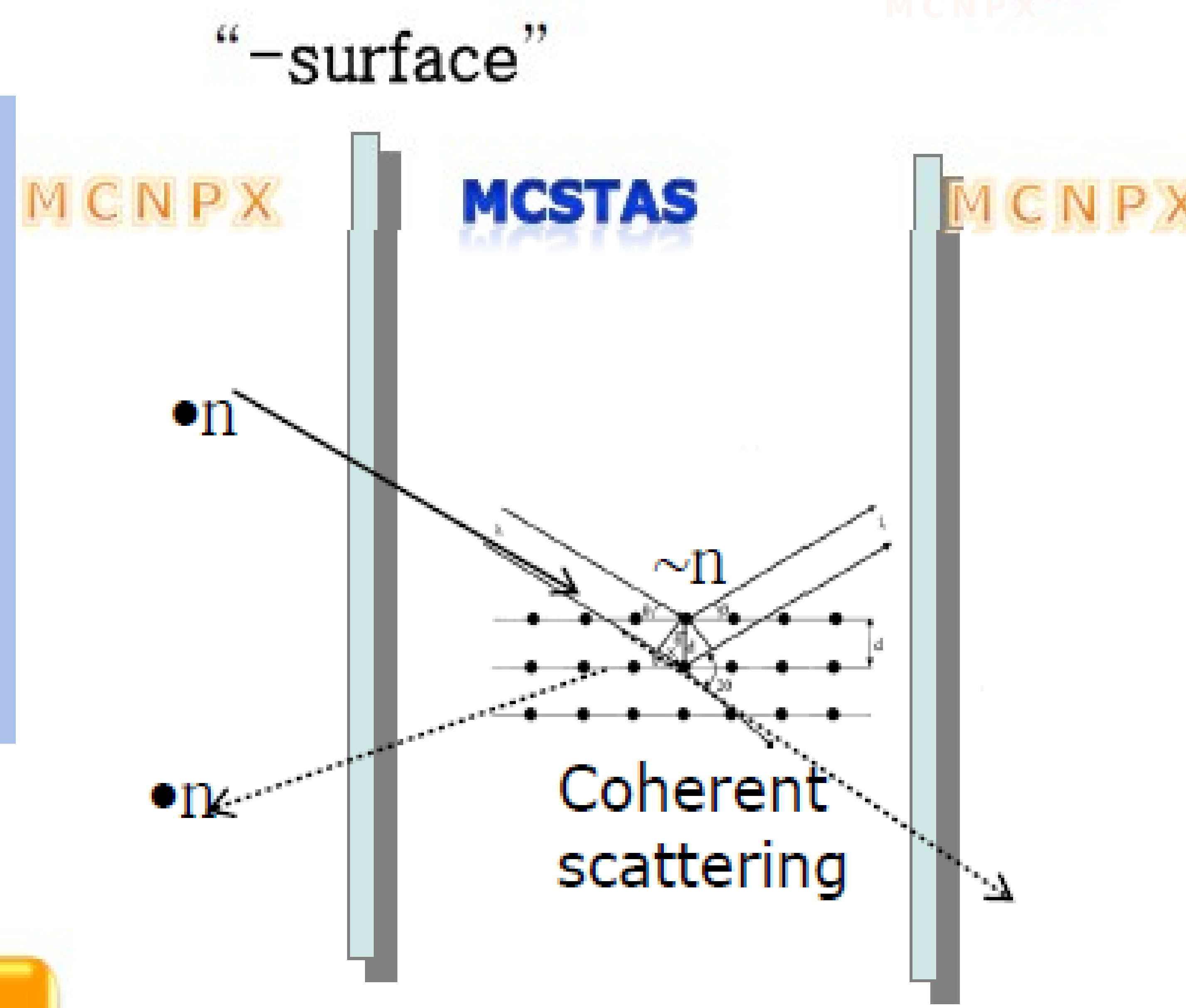




Interface option : **Combined compilation**

Method

- ➢McStas surface *flag* introduced in MCNPX
- ➢Neutron crossing McStas surface causes initiation of McStas simulation, based on neutron state.
- ➢Updated neutron state returned to MCNPX



- •Technically difficult to make general
- • Licensing issue
- • Slow: MCNPX called for each neutron

- •Potentially very flexible (but not yet fully developed)
- • All McStas functionality usable
- • Re-entry supported
- • Correlations conserved (e.g. E,pos)

<u>in MCNPX input file:</u>

…..

-110 PX -0.2

-120 PX -0.4

Interface option : **Supermirror**

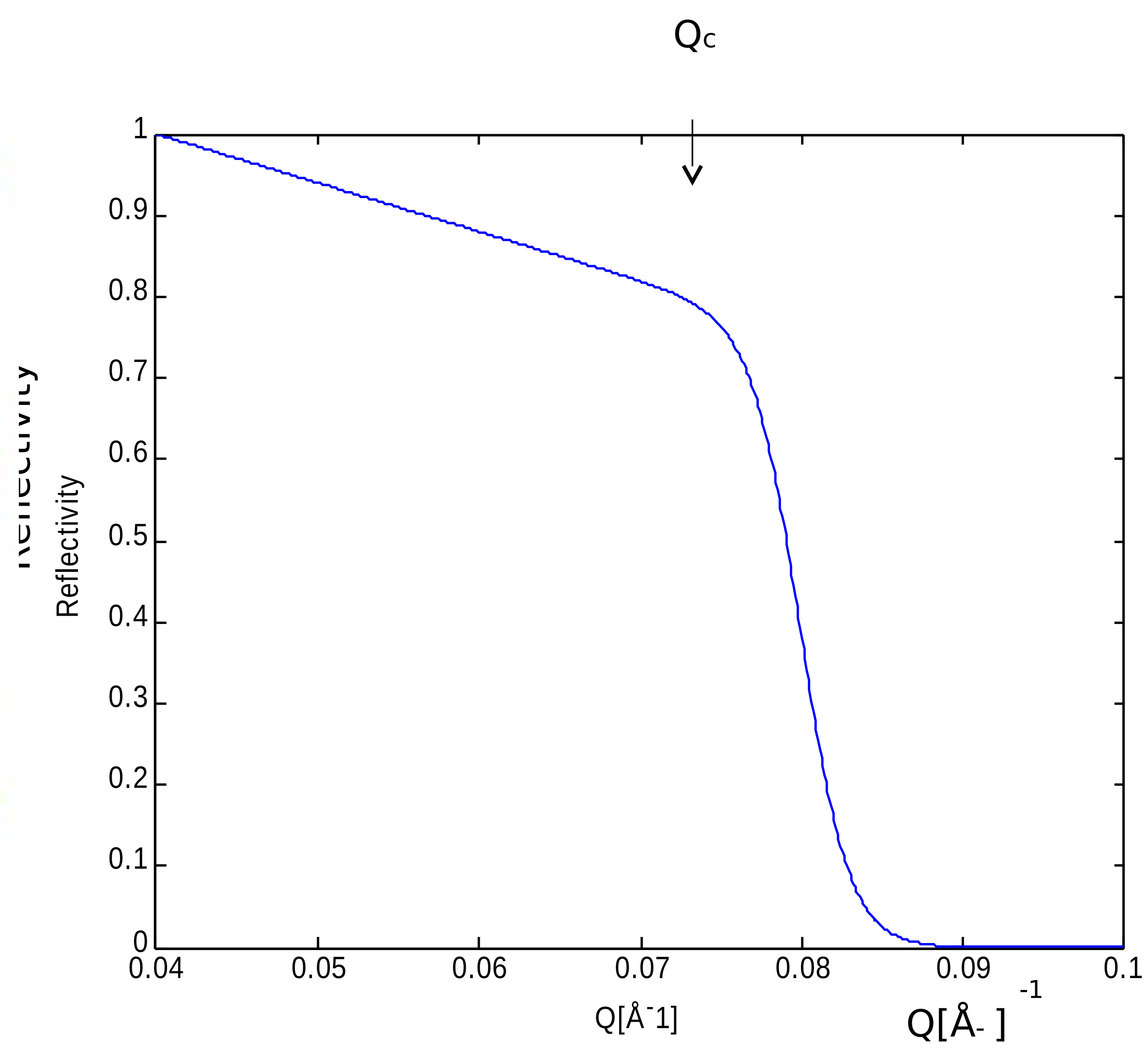
- Existing implementation, introducing McStas inspired supermirrors as a surface card in MCNPX (Gallmeier et all, Nuc.Tech. 168(3))

- Reflectivity $R=R0$ if $Q<Qc$
- $R=R0/2\{1-\tanh[(Q-mQc)/W]\}\{1-a(Q-Qc)\}$ if $Q>Qc$

- Ported to MCNPX 2.7

- Doesn't scale: workload per functionality significant. Only McStas mirrors ported
  - Licensing issue

- Re-entry supported
  - Correlations conserved (e.g. E,pos)
  - Avoids intermediate files and multiple codes

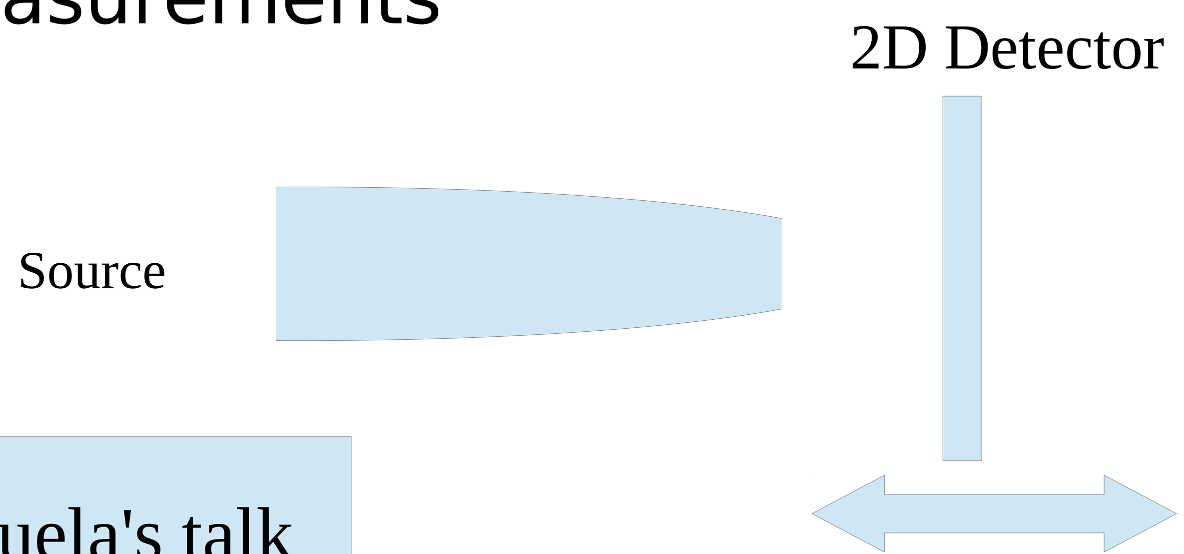


Interface applications: Performance of adaptive optics

- Similar setup as before: test beam profiles for different guide geometries
  - 1m parabolic, focusing lens (scan: 200mm-600mm along beam)

Measurements

Simulations



See Emmanouela's talk

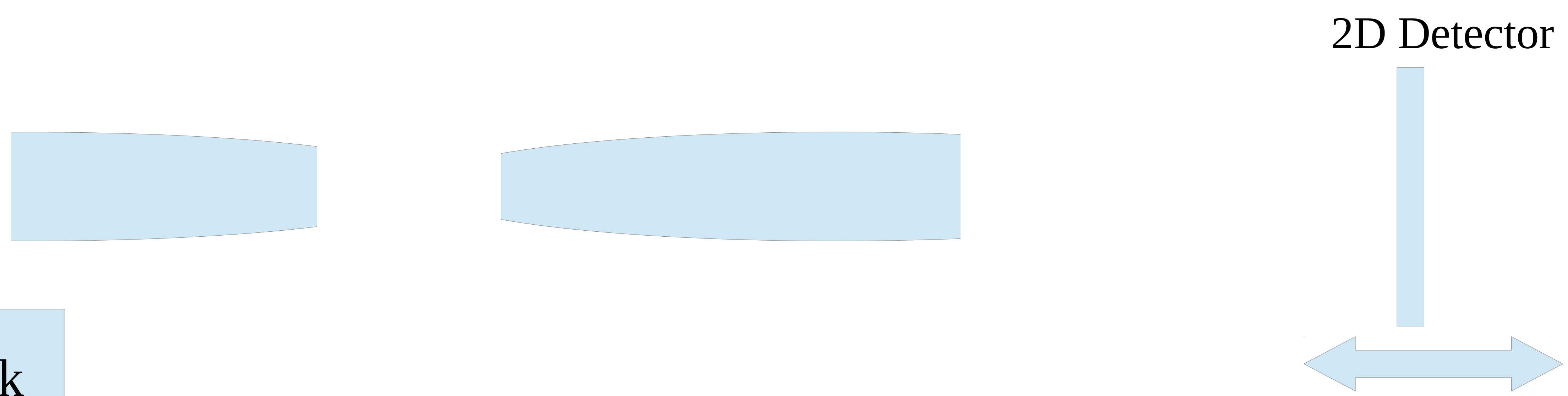
Interface applications: Performance of adaptive optics

- Similar setup as before: test beam profiles for different guide geometries
  - 1m parabolic, focusing lens + 0.5m parabolic, defocusing lens (scan:300-2000)

Measurements

Simulations

Source



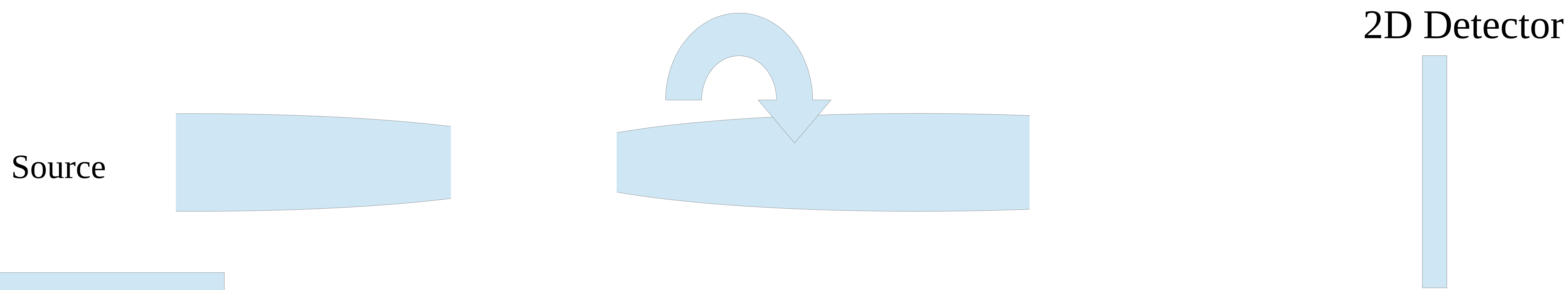
See Emmanouela's talk

Interface applications: Performance of adaptive optics

- Similar setup as before: test beam profiles for different guide geometries
  - Both lenses are mounted, and 300mm after the second lens's exit, the CCD. Second lens rotated ±0.04º in steps on 0.001º,

Measurements

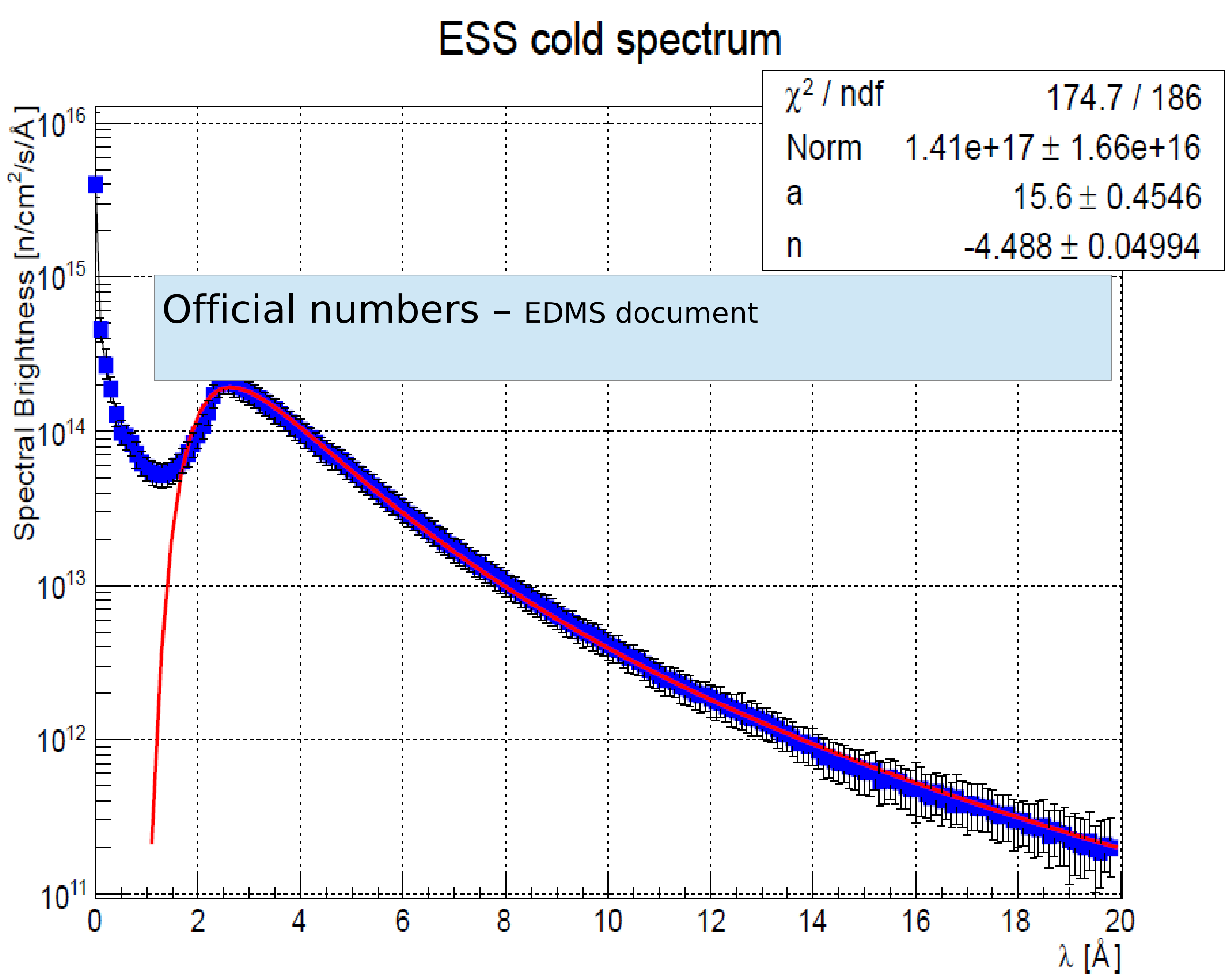
Simulations

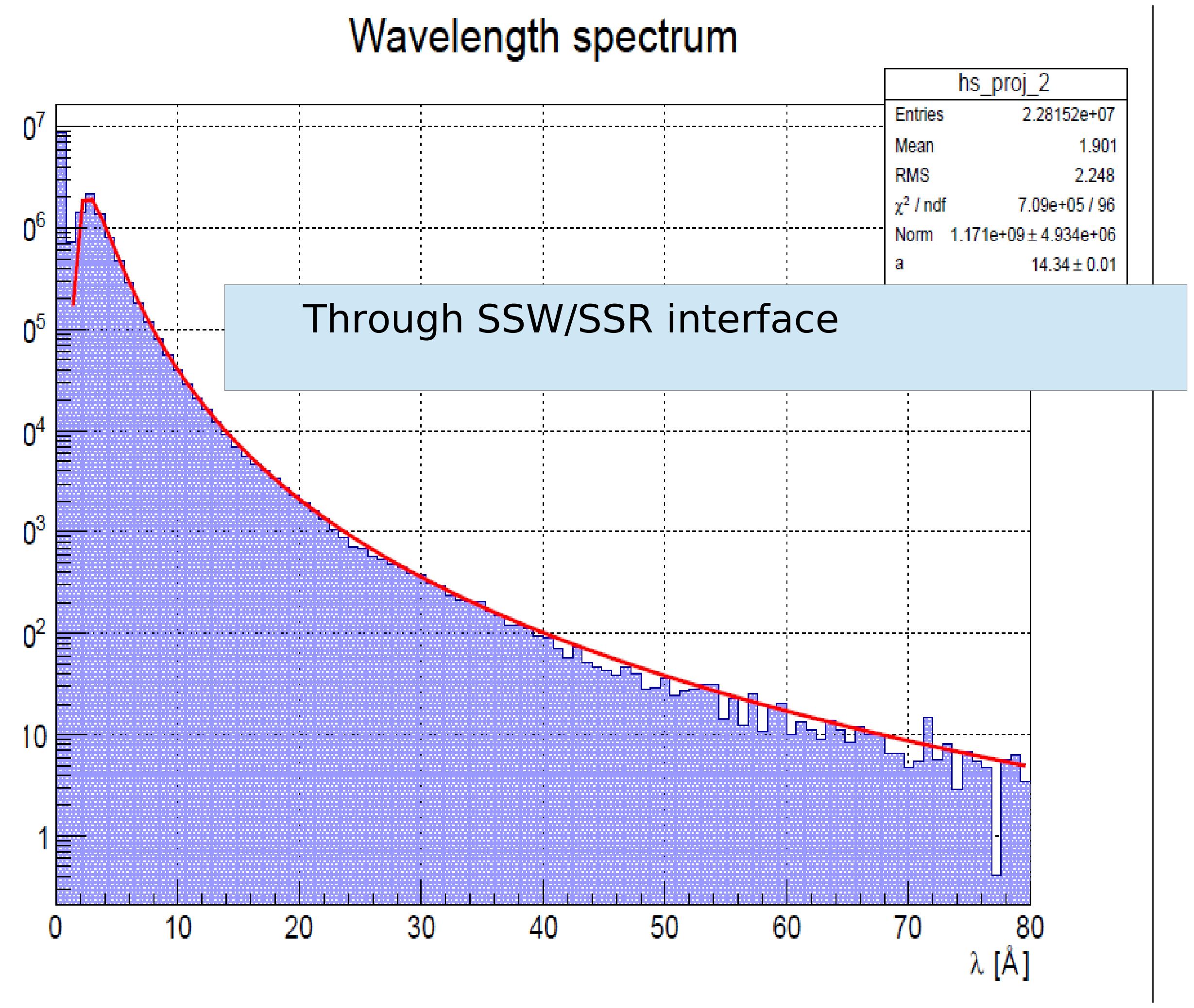


See Emmanouela's talk

McStas ESS update 1: revision of existing ESS source

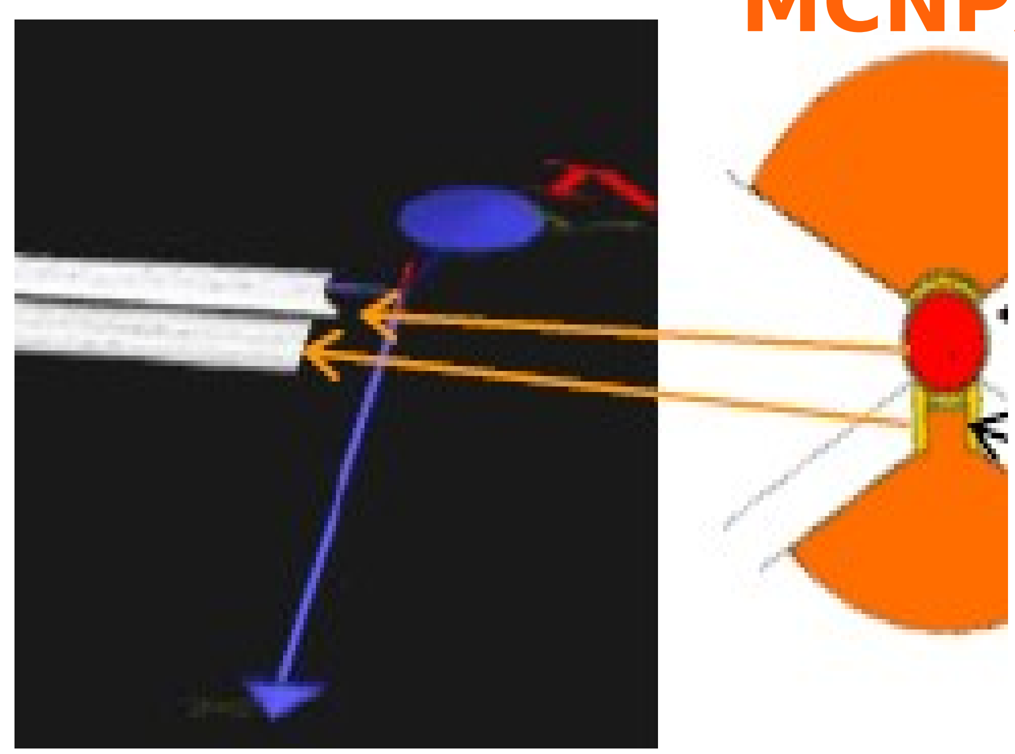
- Aimed “Post-TDR”, but good to have ~Jan 1st





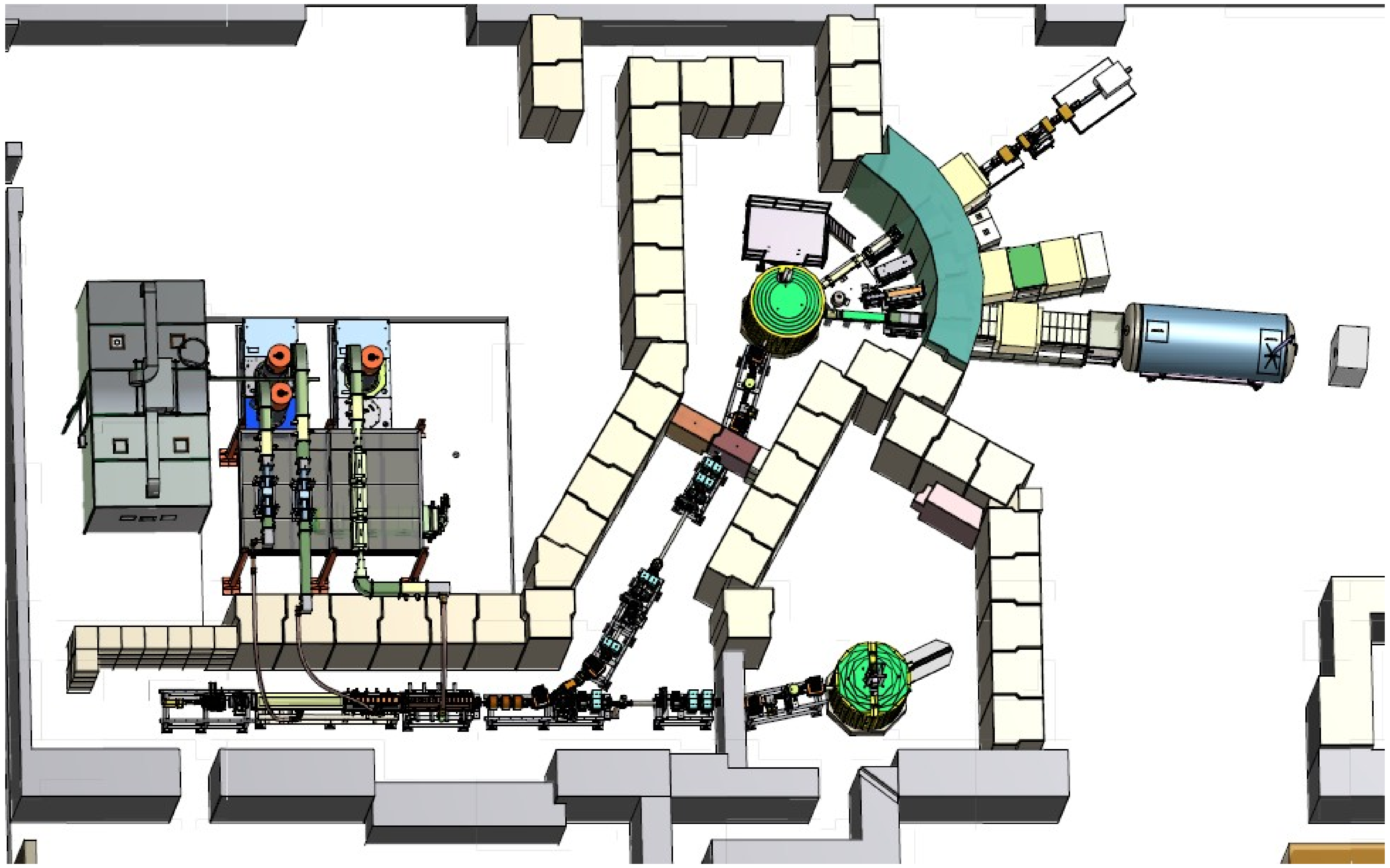
- Fitted using: $$B(\lambda,T)=Norm*\lambda^{n}*\exp(-a/\lambda^{2})$$
- Parameters *a*, l similar.
- → Cold [Ö]
- → Thermal
  Should we rather stick with the old numbers
  (ancient: Maxwellian 325K,...)
- Geometry [Ö]

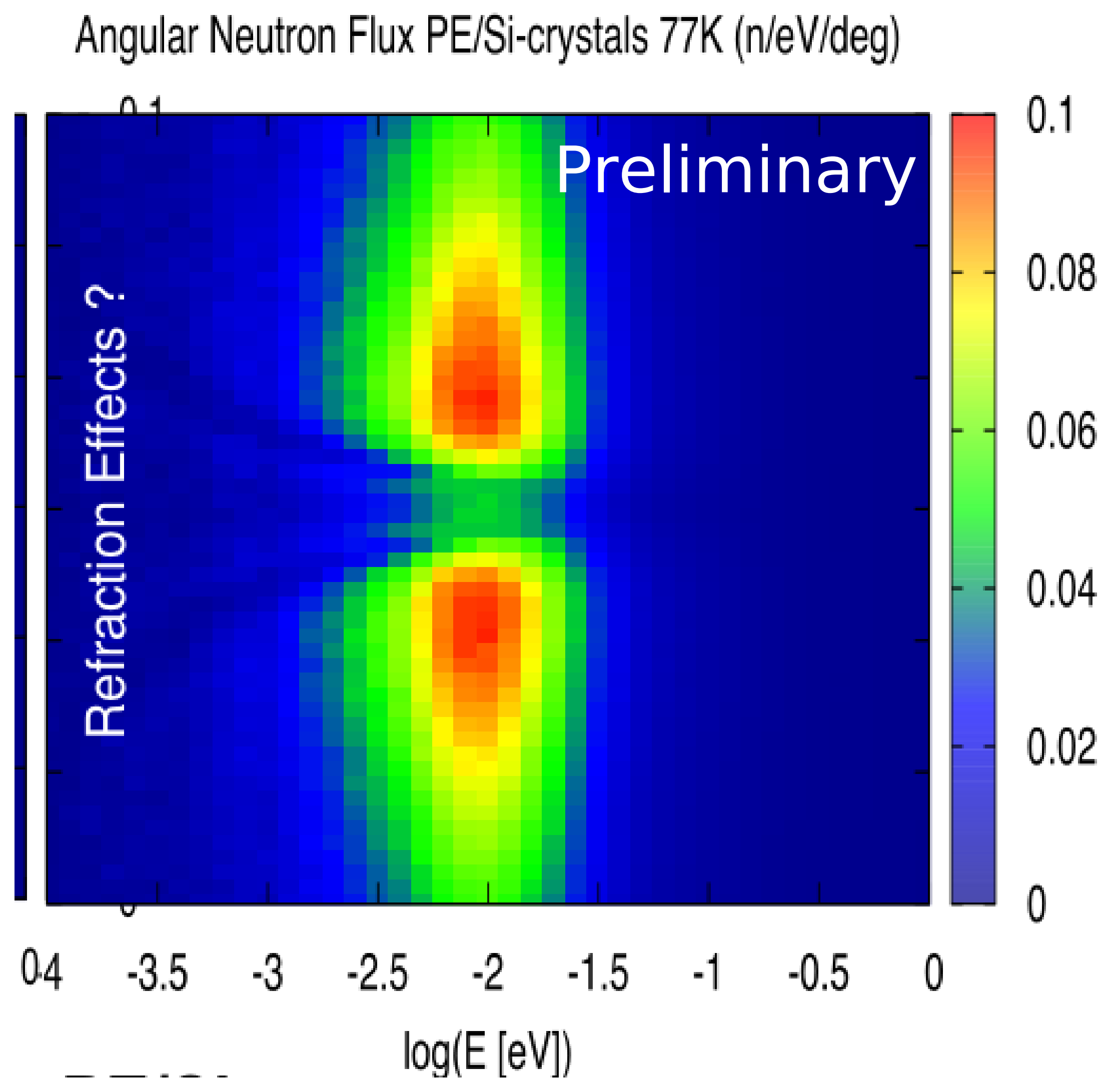
McStas **MCNPX**



Interface option : Example: **Single crystals**

- Collaborators from LENS / SNS / Los Alamos are using this approach for studying moderators





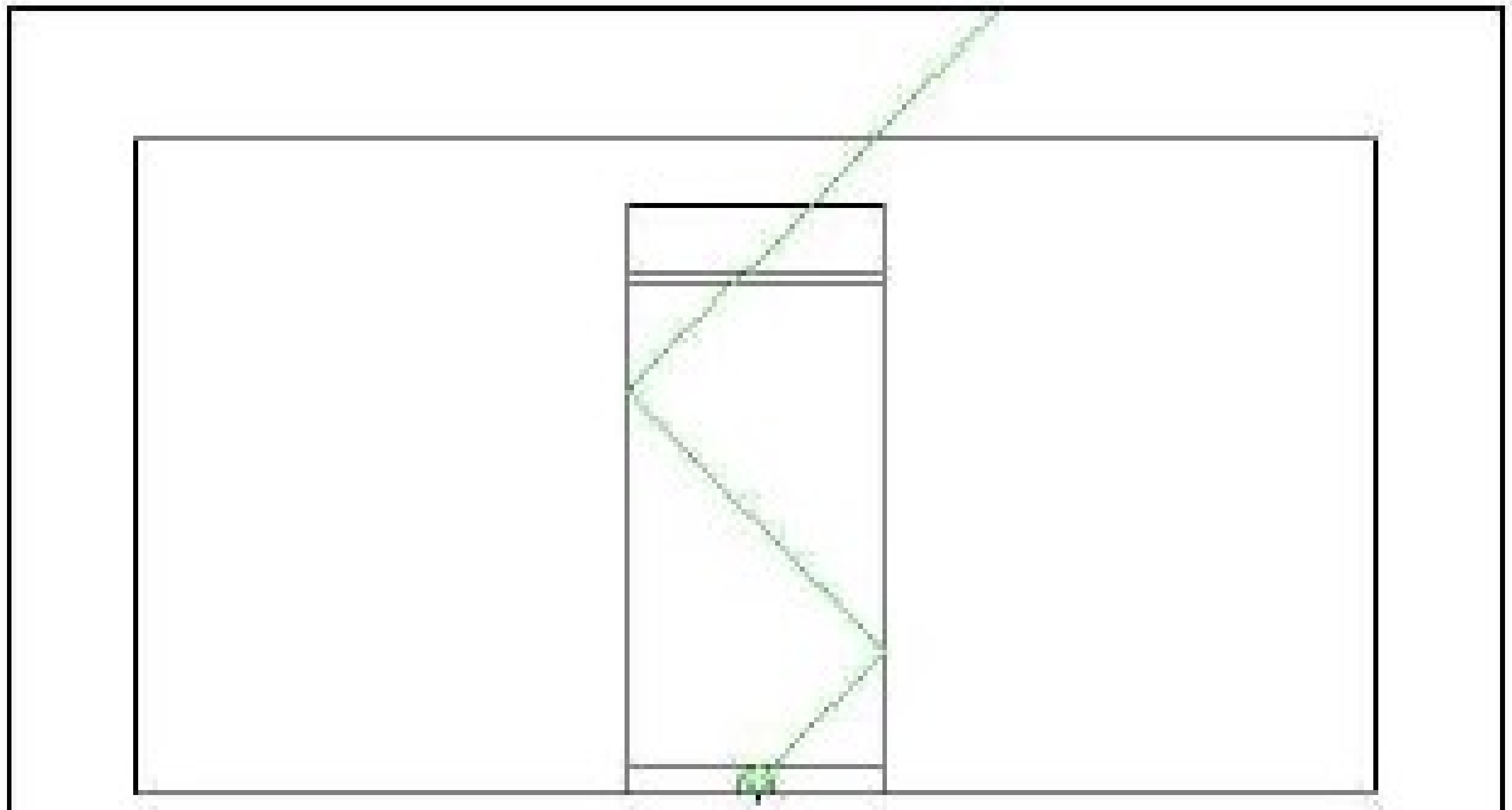
Single crystal in MCNPX - preliminary

Reminder

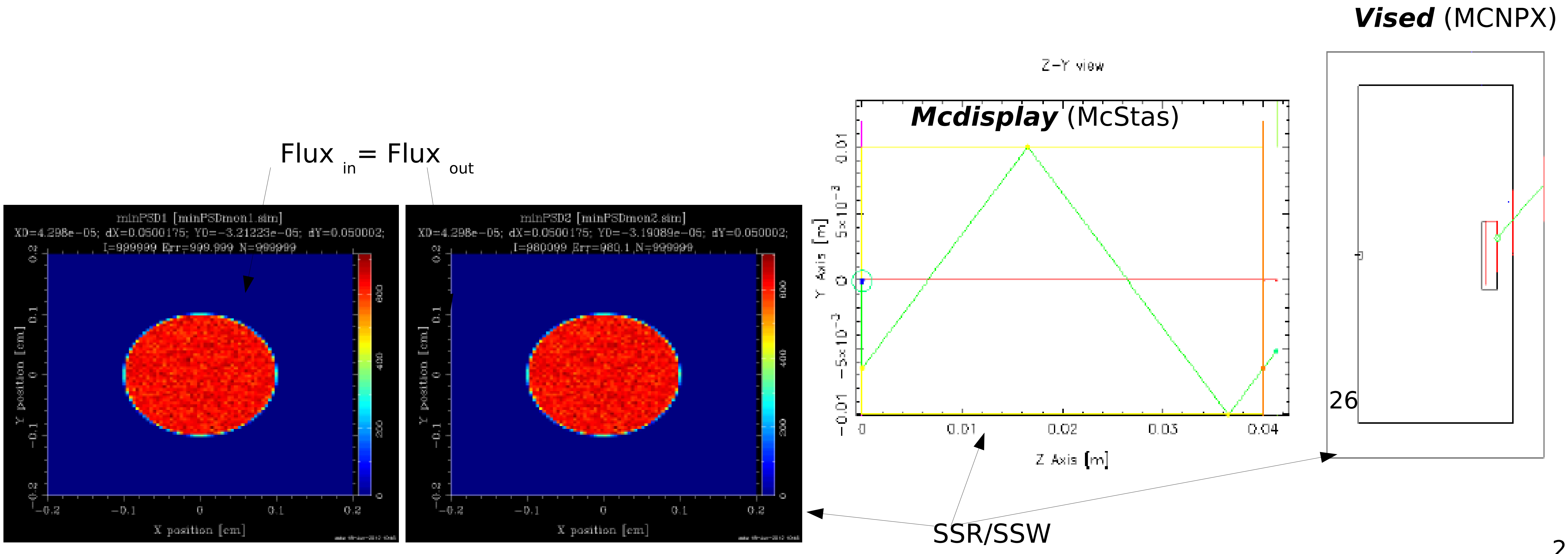
Validation setup

Strategy: consider dummy geometry, where the correct result is obvious:

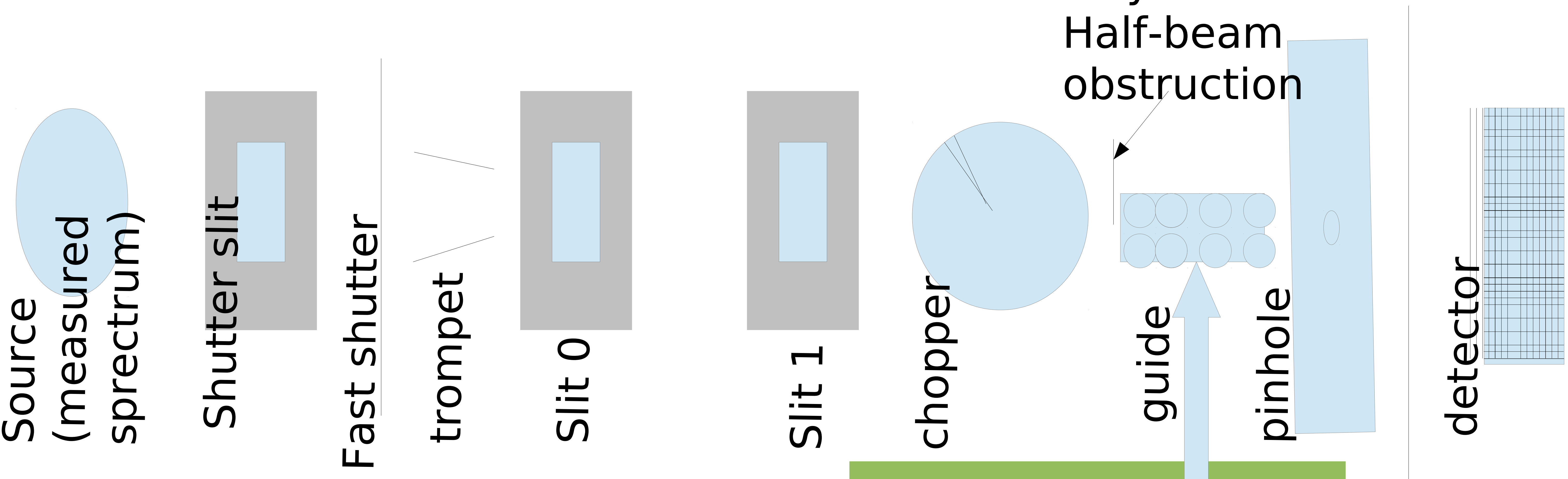
- 20meV neutrons generated at disk and aimed 45 degree toward a perfectly reflecting 'guide wall' 1 cm away (in y)
- At z=4cm: check what comes through
- Assume vacuum in guide so that transport in McStas MCNPX should be identical



MCNPX – perfect mirror



Downstream material: BOA–from cradle to grave

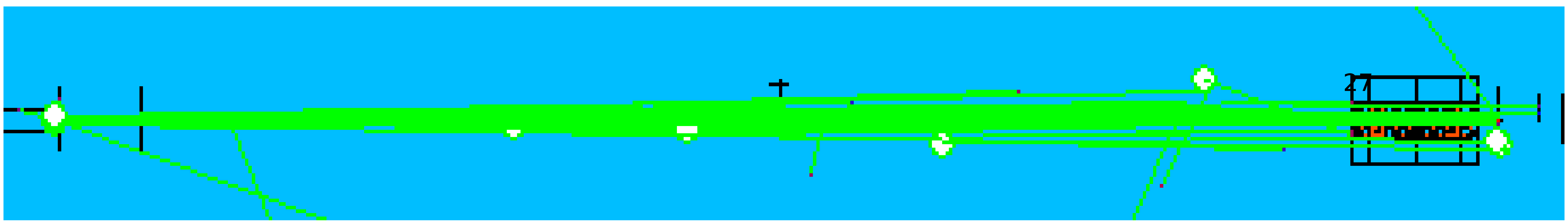


Using supermirror
In MCNP interface

Disclamer: shutter slit
and chopper are missing
In the mcnpx simulation

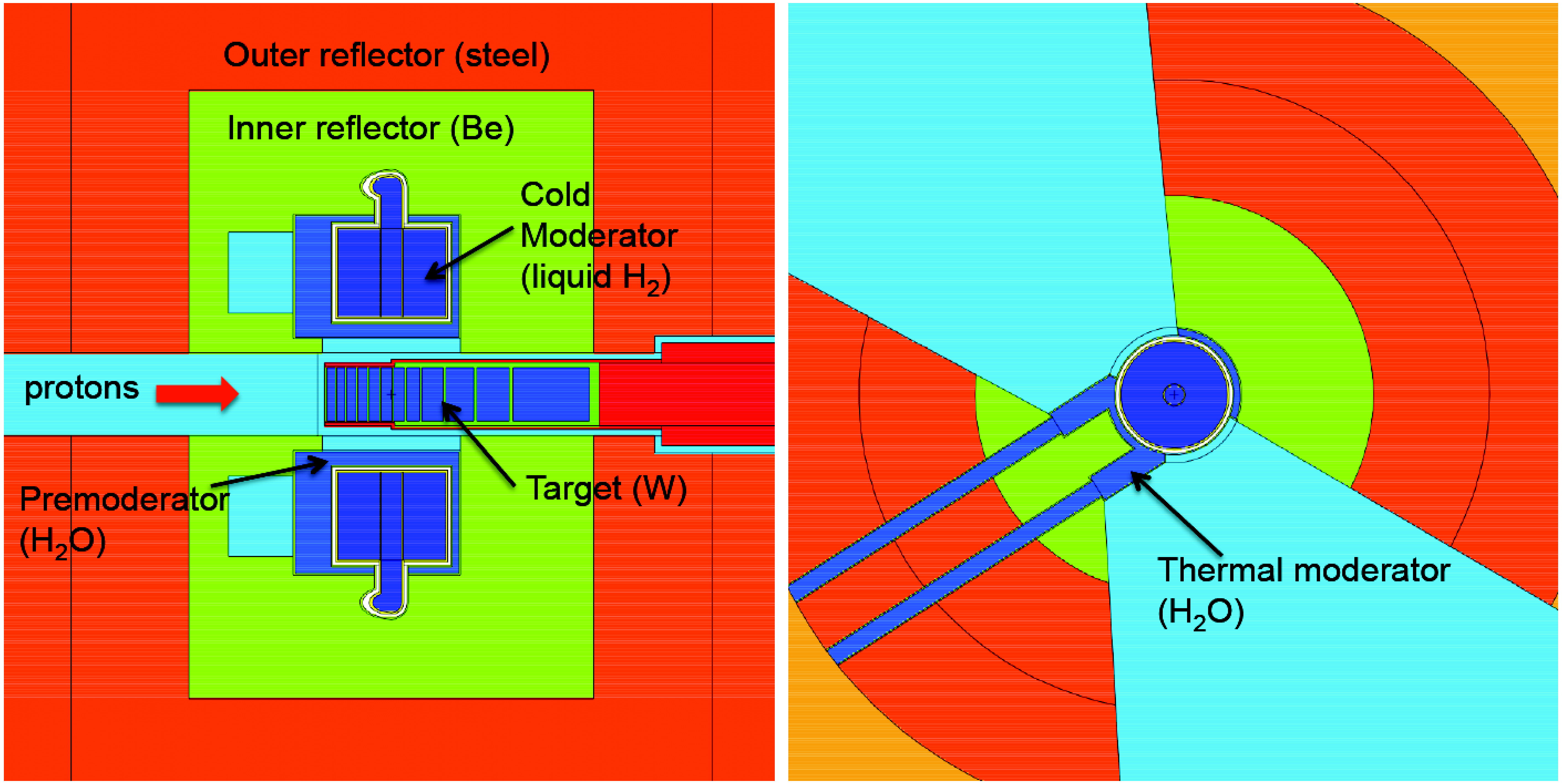
End of MCNPX sim
→ start of McStas
Through SSW interface

From MCNP Vised



ESS design concepts

- Scale of interest is ~nucleus-nucleus separation.
- ⇒ Neutrons with λ ~1-10Å are useful (~20meV: solid excitation scale)
- ⇒ Moderation needed. Choice: H_2O for thermal and liquid H_2 at 47K



Scatter logger

Components for logging scatter events and monitoring their side effects

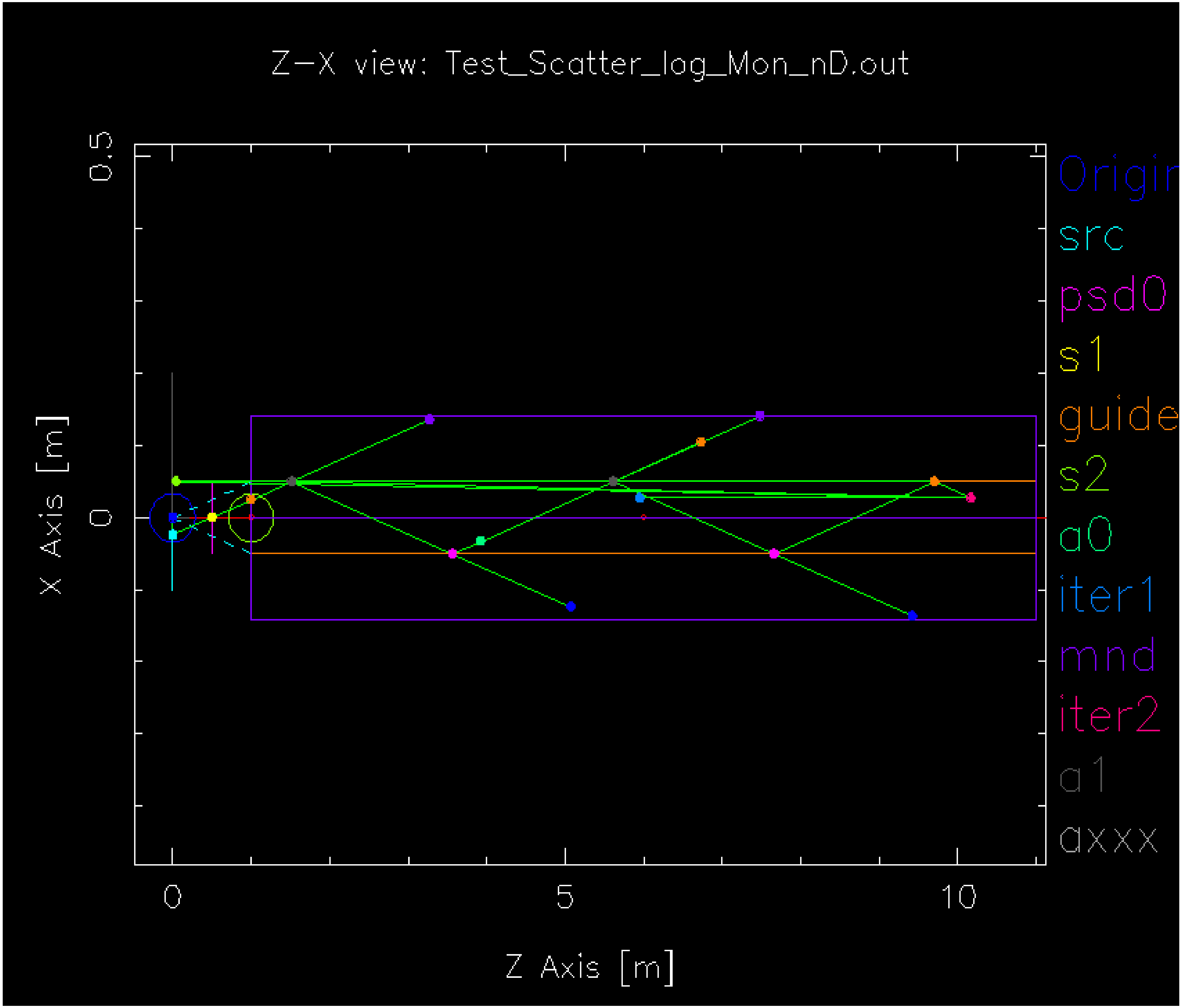
- Scatter_logger.comp
- Scatter_logger_stop.comp
- Scatter_log_iterator.comp
- Scatter_log_iterator_stop.comp

Front-end: logger

```
COMPONENT src = Source_simple(
    radius = 0.1, dist = 1, focus_xw = 0.1, focus_yh = 0.1, lambda0=5, dlambda=4.9)
AT (0, 0, 0) RELATIVE Origin

COMPONENT psd0=PSD_monitor(
    xwidth=0.1, yheight=0.1, filename="psd0")
AT(0,0,0.5) RELATIVE PREVIOUS

COMPONENT s1=Scatter_logger()
AT(0,0,1) RELATIVE src

COMPONENT guide_simple = Guide(
    w1 = 0.1, h1 = 0.1, w2 = 0.1, h2 = 0.1, l = 10, R0 = 0.99,
    Qc = 0.0219, alpha = 6.07, m = 2, W = 0.003)
AT (0, 0, 1) RELATIVE src

COMPONENT s2=Scatter_logger_stop(logger=s1)
AT(0,0,0) RELATIVE PREVIOUS
```

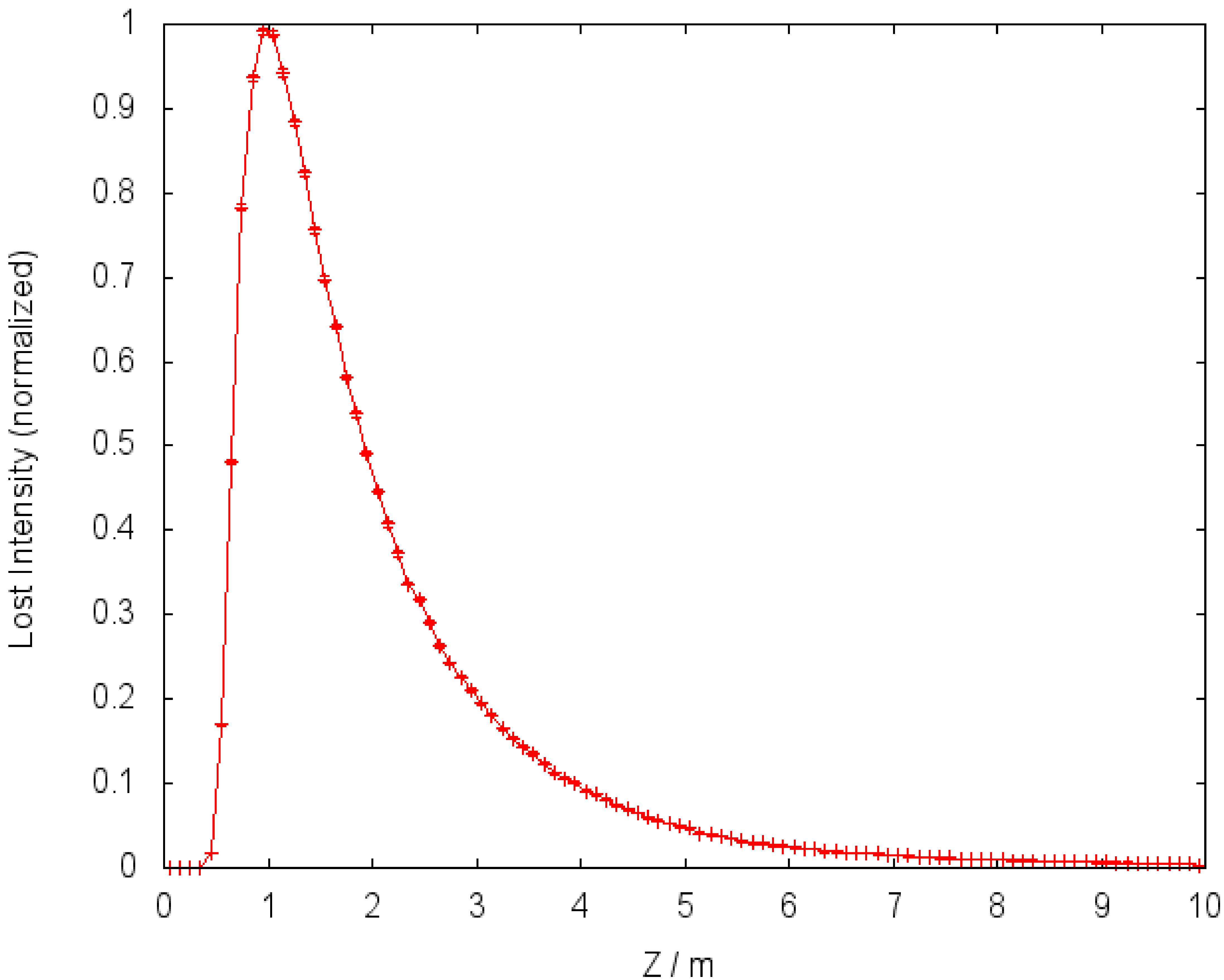
Back-end: logger iterator

```
COMPONENT a0=Arm()
AT(0,0,0) ABSOLUTE

COMPONENT iter1 = Scatter_log_iterator()
AT(0,0,0) ABSOLUTE

COMPONENT mnd=Monitor_nD (
    restore_neutron=1, yheight=10, radius=M_SQRT2*0.1,
    options="previous no slit y bins=100", filename="mnd1.dat")
AT(0,0,5) RELATIVE guide_simple
ROTATED (90,0,0) RELATIVE guide_simple

COMPONENT iter2 = Scatter_log_iterator_stop(iterator=iter1)
AT(0,0,0) RELATIVE iter1

COMPONENT a1 = Arm()
AT (0,0,0) ABSOLUTE
JUMP a0 WHEN(MC_GETPAR(iter2,loop))
```



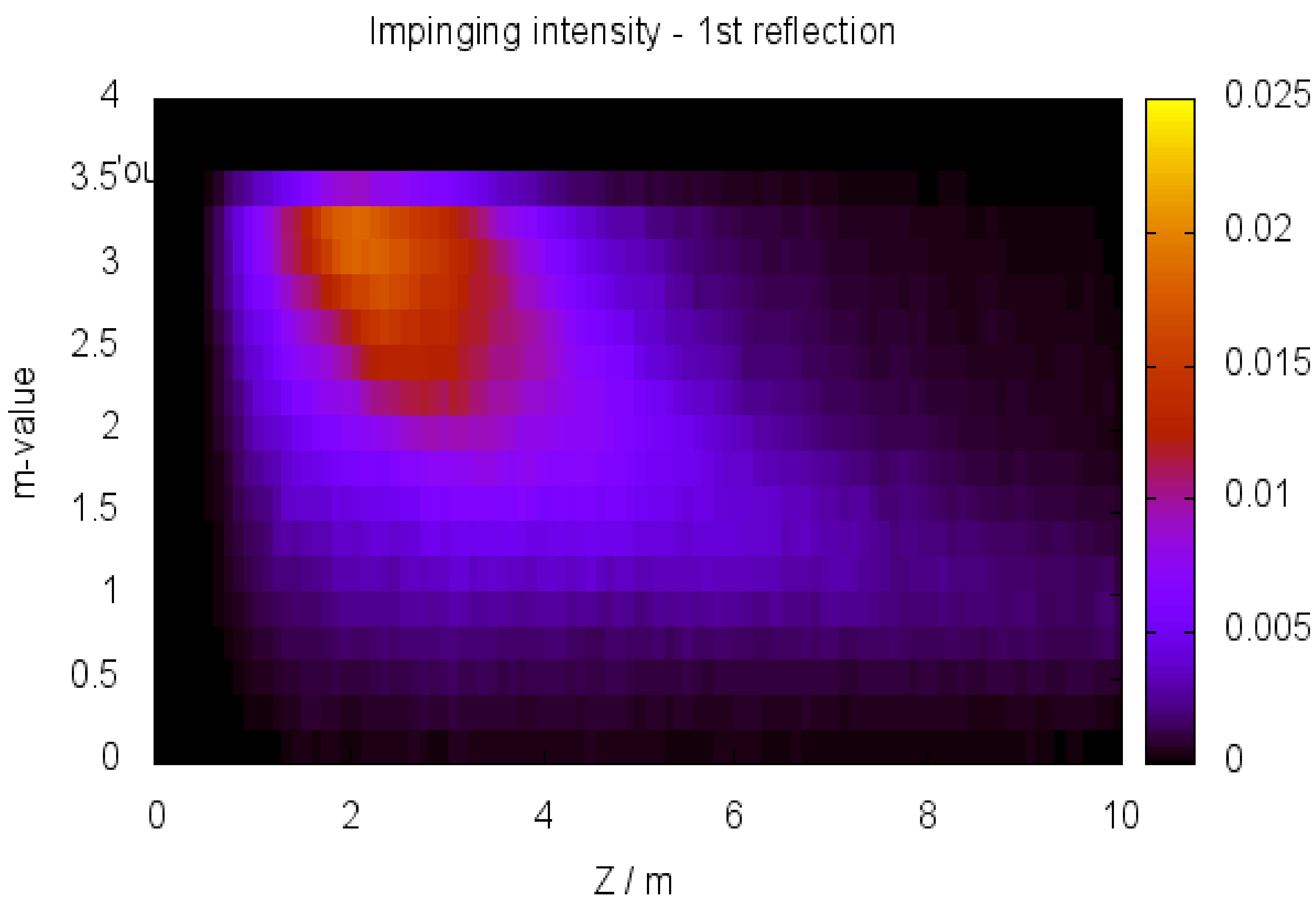
Z-X view: Test_Scatter_log_Mon_nD.out
X Axis [m]
Z Axis [m]
0.5
0
0
5
10
Origin
src
psd0
s1
guide
s2
a0
iter1
mnd
iter2
a1
axxx

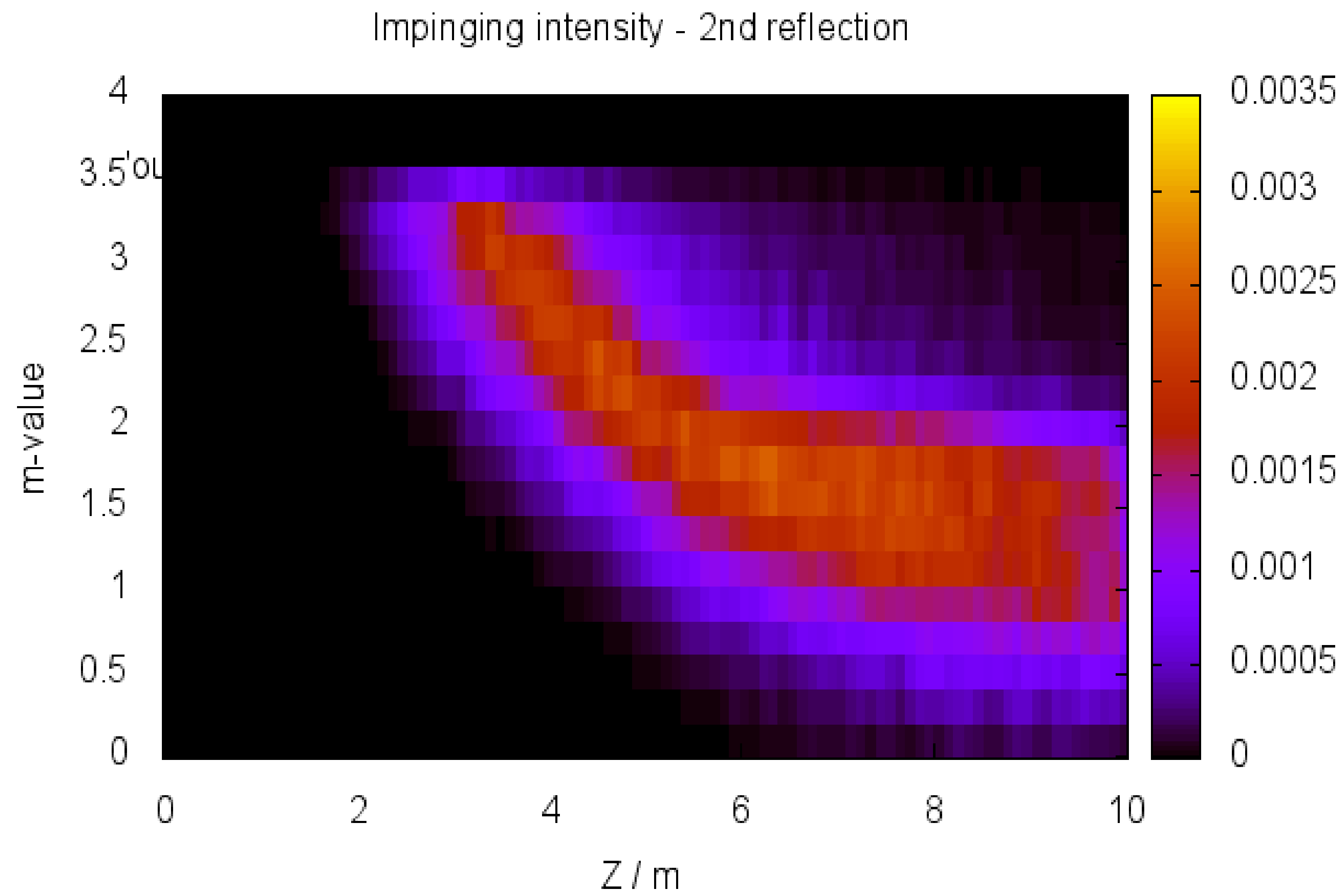
Intensity lost along guide

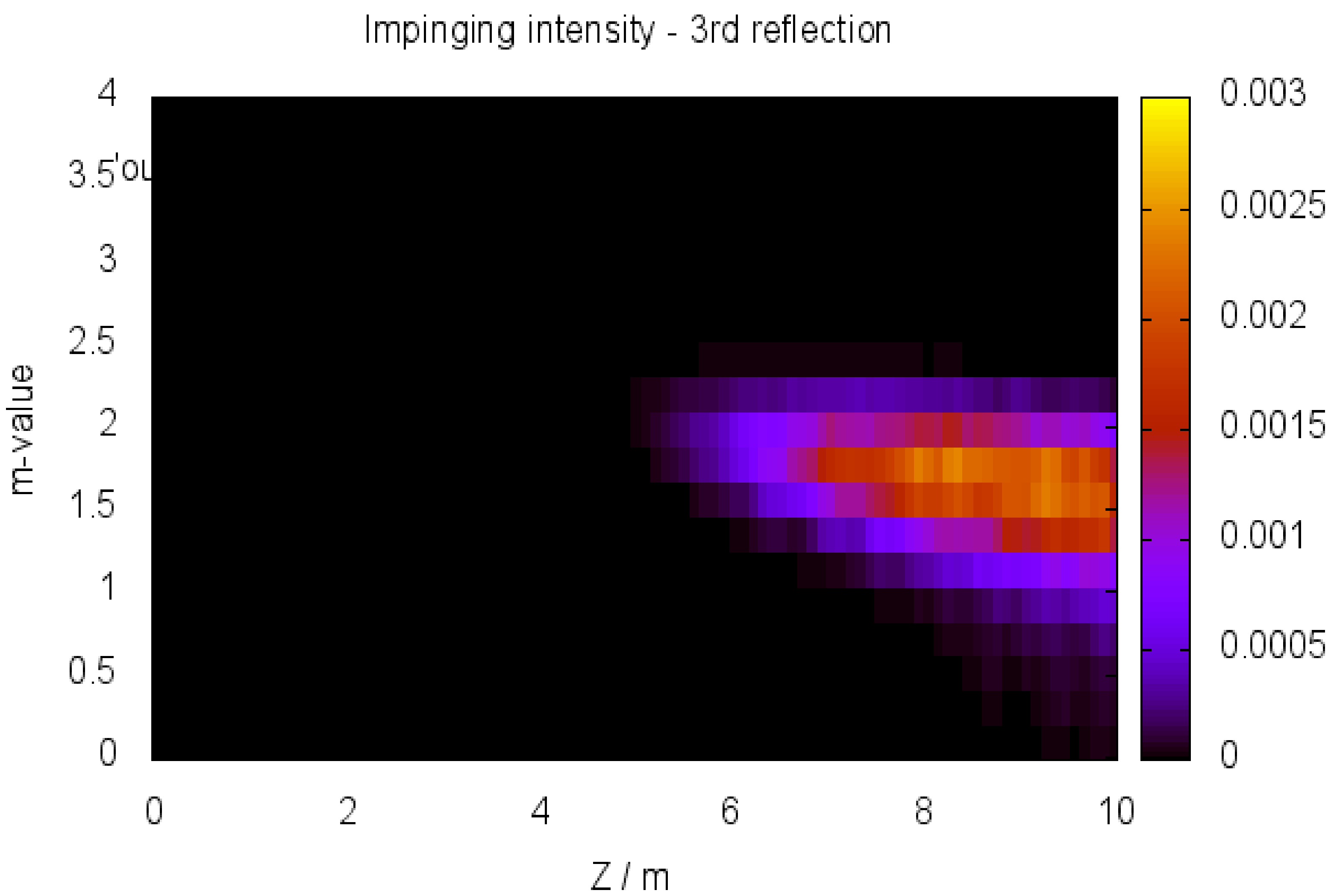


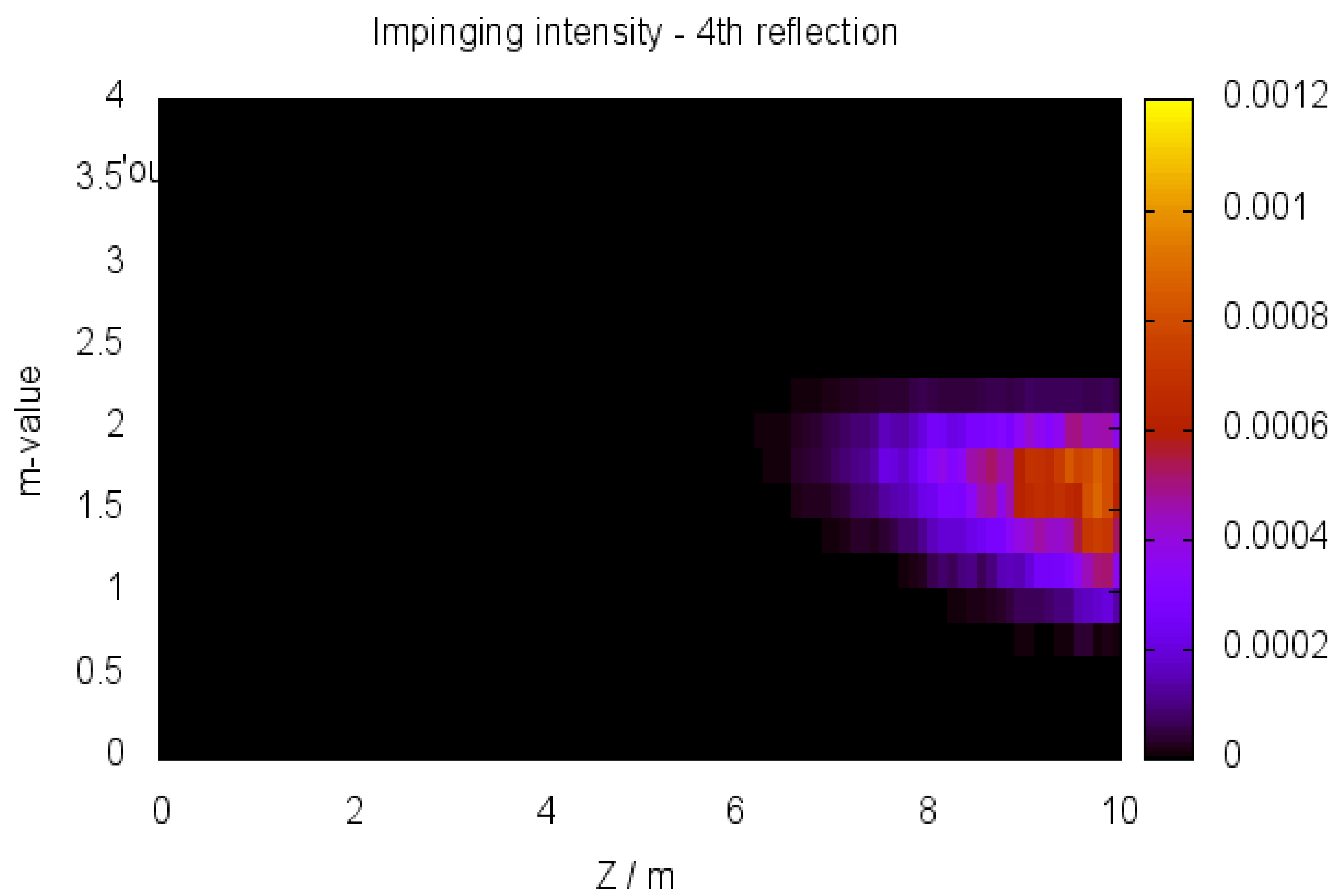
Specialized pseudo neutron state function

```
double mvalue;

int reflc;

int reflect_m-value(double *ns_tilde, struct Generalized_State_t *S0, struct Generalized_State_t *S1){

    /*position comes from "new" state*/

    ns_tilde[0]=S1->_x;ns_tilde[1]=S1->_y;ns_tilde[2]=S1->_z;

    /*velocity is the "old" state*/

    ns_tilde[3]=S0->_vx;ns_tilde[4]=S0->_vy;ns_tilde[5]=S0->_vz;

    /*time from new*/

    ns_tilde[6]=S1->_t;

    /*weight is impinging weight - old state*/

    ns_tilde[10]=S0->_p;



    double v = sqrt(S0->_vx*S0->_vx+S0->_vy*S0->_vy+S0->_vz*S0->_vz);

    double k = v*V2K;

    double theta  = 0.5*acos(scalar_prod(S0->_vx,S0->_vy,S0->_vz,S1->_vx,S1->_vy,S1->_vz)/(v*v));

    mvalue = 2*k*sin(theta)/0.0219; //EK doublecheck

    reflc=S1->_idx;

    return 0;

}
```


Impinging intensity - 1st reflection
m-value
Z / m
0 0.5 1 1.5 2 2.5 3 3.5'ou 4
0 2 4 6 8 10
0 0.005 0.01 0.015 0.02 0.025


Impinging intensity - 2nd reflection
m-value
Z / m
0 0.5 1 1.5 2 2.5 3 3.5'ou 4
0 2 4 6 8 10
0 0.0005 0.001 0.0015 0.002 0.0025 0.003 0.0035


Impinging intensity - 3rd reflection
m-value
Z / m
0 0.5 1 1.5 2 2.5 3 3.5'ou 4
0 2 4 6 8 10
0 0.0005 0.001 0.0015 0.002 0.0025 0.003


Impinging intensity - 4th reflection
m-value
Z / m
0 0.5 1 1.5 2 2.5 3 3.5'ou 4
0 2 4 6 8 10
0 0.0002 0.0004 0.0006 0.0008 0.001 0.0012

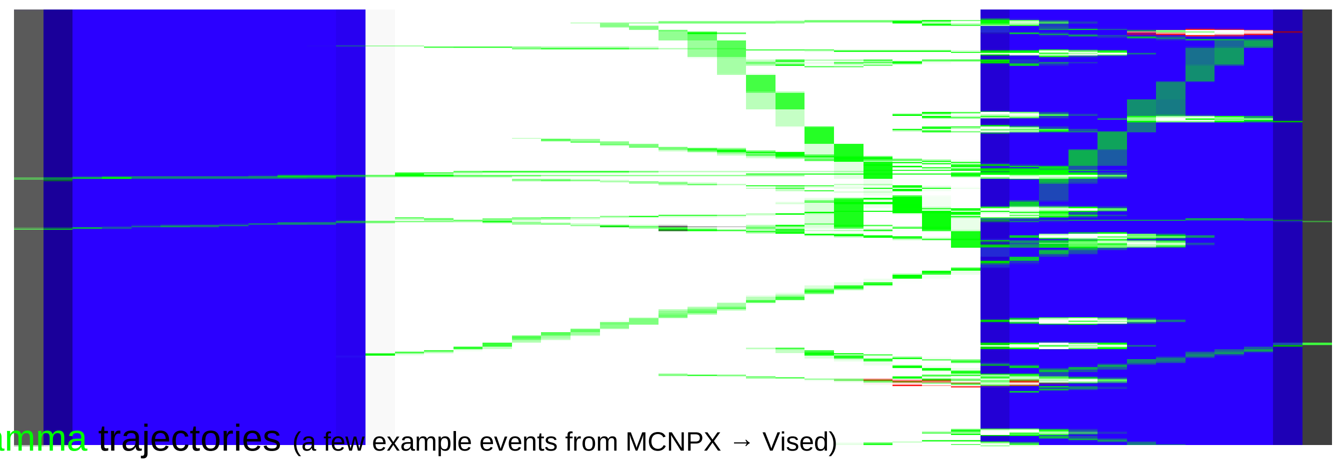
Another example: Specialized pseudo neutron state function → background along guide

```
/*position comes from "new" state*/
ns_tilde[0]=S1->_x;ns_tilde[1]=S1->_y;ns_tilde[2]=S1->_z;

/*velocity is the "old" state*/
ns_tilde[3]=S0->_vx;ns_tilde[4]=S0->_vy;ns_tilde[5]=S0->_vz;

/*time from new*/
ns_tilde[6]=S1->_t;

/*weight is difference old-new to mean the neutrons "deposited" in the guide wall*/
ns_tilde[10]=S0->_p-S1->_p;
```

Same as before

I.e.: The temporarily stored state is the ***un-reflected neutrons*** - normally discarded

Using the MCNPX-McStas interface: *Virtual_MCNP_ss_output.comp* (McStas 2.0), the simulation of absorbed neutrons proceeds:



Neutron and gamma trajectories (a few example events from MCNPX → Vised)

Next: use to calculate shielding requirements for realistic ESS guide geometry and source

Flip-view